ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏ

Բունիաթյան Դավիթ Գագիկի

Կենսաբժշկական պատկերների խորը ուսուցման տարաբաշխված ամպային մեթոդների մշակում

ՍԵՂՄԱԳԻՐ

Ե.13.04 «Հաշվողական մեքենաների, համալիրների, համակարգերի և ցանցերի մաթեմատիկական և ծրագրային ապահովում» մասնագիտությամբ տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի համար

Երևան - 2020

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ НАН РА

Буниатян Давит Гагикович

Разработка распределенных облачных методов глубокого обучения для биомедицинских изображений

АВТОРЕФЕРАТ

Диссертации на соискание ученой степени кандидата технических наук по специальности: 05.13.04 – математическое и программное обеспечение вычислительных машин, комплексов, систем и сетей

Ереван - 2020

Ատենախոսության թեման հաստատվել է Հայ-Ռուսական համալսարանում:

Գիտական ղեկավար՝	ֆիզ.-մաթ. գիտ. դոկտոր Հ . Գ. Սարուխանյան
Պաշտոնական ընդդիմախոսներ՝	ֆիզ.-մաթ. գիտ. դոկտոր Ս. Կ. Շուքուրյան
	տեխ. գիտ. թեկնածու Գ. Ա. Կարապետյան
Առաջատար կազմակերպություն՝	Հայաստանի ազգային պոլիտեխնիկական համալսարան

Պաշտպանությունը կայանալու է 2020թ. ապրիլի 10-ին, ժ. 17^{00}-ին ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում գործող 037 «Ինֆորմատիկա» մասնագիտական խորհրդի նիստում հետևյալ հասցեով՝ Երևան, 0014, Պ. Սևակի 1:

Ատենախոսությանը կարելի է ծանոթանալ ՀՀ ԳԱԱ ԻԱՊԻ գրադարանում:

Սեղմագիրն առաքված է 2020թ փետրվարի 28-ին:

Մասնագիտական խորհրդի գիտական քարտուղար, ֆիզ.-մաթ.գիտ. դոկտոր՝ Հ.Գ. Սարուխանյան

Тема диссертации утверждена в Российско-Армянском университете.

Научный руководитель:	доктор физ.-мат. наук А. Г. Саруханян
Официальные оппоненты:	доктор физ.-мат. наук С. К. Шукурян
	кандидат тех. наук Г. А. Карапетян
Ведущая организация:	Национальный политехнический университет Армении

Защита состоится 10-го апреля 2020г. в 17^{00} часов на заседании специализированного совета 037 Информатика Института проблем информатики и автоматизации НАН РА по адресу: 0014 г. Ереван, ул. П. Севака 1.

С диссертацией можно ознакомиться в библиотеке ИПИА НАН РА.

Автореферат разослан 28-го февраля 2020г.

Ученый секретарь специализированного совета, доктор физ.-мат. наук А.Г. Саруханян

ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ԸՆԴՀԱՆՈՒՐ ԲՆՈՒԹԱԳԻՐԸ

Ատենախոսության թեմայի արդիականությունը: Խոր ուսուցումը և մեծ տվյալներն ունեն բազմազան կիրառություններ. տրանսպորտում (ինքնավար մեքենաներ), գյուղատնտեսությունում (մոլախոտերի հայտնաբերում), արդյունաբերության մեջ (արդյունաբերական մեքենաների, հաստոցների սխալանքների հայտնաբերում) և կենսաբժշկության ոլորտում (չարորակ ուռուցքների ախտորոշում կամ գլխուղեղի պատկերների հետազոտություններ):

Նեյրոգիտության ոլորտում ուղեղի նեյրոնների անատոմիական գրաֆի վերակառուցումը թույլ է տալիս հետևություններ կատարել բնական նեյրոնային ցանցի հատկությունների վերաբերյալ: Կոնեկտոմը գլխուղեղի ուղղորդված մուլտիգրաֆն է, որի գագաթները նեյրոններն են, իսկ կապերը՝ սինապսները: Դրոզոֆիլիայի (ճանճի) գլխուղեղի բարձր որակի անատոմիական վերակառուցումը սկսվում է նմուշը բարակ (40 նանոմետր՝ նմ) շերտերի կտրելով, որից հետո հավաքվում են նրանց պատկերները էլեկտրոնային մանրադիտակի միջոցով: Պատկերների վերադրման (image alignment) նպատակն է՝ հավաքված շերտերի 2D պատկերներից ստանալ գլխուղեղի նմուշը ներկայացնող 3D պատկեր (հատումների քանակը՝ 7600, յուրաքանչյուր հատույթ պարունակում է 187000x92000 պիկսել, հիշողությունը՝ 100ՏԲ)[1]: Վերադրումից հետո առանձնացվում է յուրաքանչյուր նեյրոն՝ կիրառելով սեգմենտավորման մոդելներ: Այնուհետև հայտնաբերվում են սինապսներն ու վերակառուցվում է մուլտիգրաֆը: Խոր նեյրոնային մոդելները օգտագործվում են յուրաքանչյուր փուլում:

Վերջին մի քանի տարիների ընթացքում սեգմենտավորման մոդելները հասել են գերմարդկային ճշտության[2]: Արդյունքում գրաֆի կառուցման սխալմունքի 70%-ը գալիս է վերադրման փուլում սխալ համընկումներից[3]: Ցանկացած մեթոդի կատարելագործում նախնական փուլերում ունի թիթեռի էֆեկտ վերջնական գրաֆի ճշտության վրա: Վերադրումը իր բնույթով անվերահսկելի խնդիր է, քանզի շերտերը կտրելուց առաջ հնարավոր չէ իմանալ հնարավոր դեֆորմացիաների կամ շեղումների մասին:

Այս աշխատանքի շրջանակներում ներկայացնում ենք թույլ վերահսկելի մետրիկ ուսուցման մեթոդ, որը բարելավում է լայնորեն գործածվող էլաստիկ վերադրման մեթոդը[4]՝ կիրառելով խոր ուսուցման մոդելներ: Էլաստիկ վերադրման ընթացքում կատարվում է մի քանի միլիարդի համընկում հաջորդական շերտերում: Մեր մեթոդը նվազեցնում է շերտերի համընկման սխալմունքը 0,11%-ից մինչև 0,05% և որոշ դեպքերում՝ 0,47%-ից 0,06%:

Վերջին տարիներին մշակվել են մեծածավալ տվյալների շտեմարաններ՝ հասնելով 100 ՏԲ և նույնիսկ 1 ՊԲ (1048576 ԳԲ) ծավալի ոչ միայն նեյրոգիտության ոլորտում, այլև բնական լեզվի մշակման կամ տեսաձայնային այլ ոլորտներում: Ամպային լուծումներում 1 ՊԲ հասնող ծավալային շտեմարանների մշակումը հասնում է միլիոնավոր դոլարներին

[1]White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. (1986). The structure of thenervous system of the nematode caenorhabditis elegans. Philos Trans R Soc Lond B Biol Sci, 314(1165):1–340

[2]Lee, Kisuk, Jonathan Zung, Peter Li, Viren Jain, and H. Sebastian Seung. "Superhuman accuracy on the SNEMI3D connectomics challenge." arXiv preprint arXiv:1706.00120 (2017).

[3]Zung, Jonathan, Ignacio Tartavull, Kisuk Lee, and H. Sebastian Seung. "An error detection and correction framework for connectomics." In Advances in Neural Information Processing Systems, pp. 6818-6829. 2017

[4]Saalfeld, Stephan, Richard Fetter, Albert Cardona, and Pavel Tomancak. "Elastic volume reconstruction from series of ultra-thin microscopy sections." Nature methods 9, no. 7 (2012): 717.

համարժեք հաշվարկումների: Հաշվարկների ցանկացած օպտիմիզացիան խնայում է զգալի գումարներ: U-Net և նրա տարբերակների խոր մոդելները գրանցում են լավագույն արդյունքներ կենսաբժշկության ոլորտի խնդիրներում[5]: Այս խոր ուսուցման մոդելները արագ են աշխատում GPU-ների վրա, սակայն, հաշվի առնելով CPU մեքենաների գնային առաջարկը, մոդելների CPU հաշվարկումները կատարելագործելու դեպքում հնարավոր է խնայել զգալի ծախսեր:

Այս աշխատանքի շրջանակներում կատարելագործում ենք ծավալային պատկերների խոր մոդելների մշակման հաշվարկները Intel Xeon պրոցեսորների համար: Արդյունքում արագացնում ենք հաշվարկները 3-4 անգամ հայտնի խոր ուսուցման գրադարանների համեմատ և ստանում 1,5 անգամ ավելի էժան հաշվարկներ GPU-ների համեմատ:

Պետաբայթ ծավալի հասնող տվյալների արդյունավետ պահպանումը և տեղաբաշխումը ներկայացնում են ենթակառուցվածքային մարտահրավերներ: MapReduce[6] պարադիգմի հիման վրա բաշխված համակարգերը, ինչպես օրինակ՝ Hadoop կամ Spark[7], թերանում են խոր ուսուցման հաշվարկներ կատարելիս հատկապես ծավալային պատկերների համար:

Սույն աշխատանքում նաև ներկայացնում ենք ենթակառուցվածքային համակարգ տվյալների կենտրոնացած պահպանման և հազարավոր մեքենաների արդյունավետ կառավարման համար ամպային լուծումներում: Այն մասնագիտանում է խոր ուսուցման լայնամասշտաբ խնդիրների լուծման համար:

Միավորելով սույն աշխատանքի երեք բաղադրիչները՝ վերադրման համընկման մեթոդը, խոր ուսուցման գրադարանը և ենթակառուցվածքային համակարգը, կատարելագործում ենք պետաբայթի հասնող կենսաբժշկական մեծ պատկերների արդյունավետ վերադրումը ամպային լուծումներում: Ենթակառուցվածքը և գրադարանը նաև կարող են օգտագործվել նեյրոնների գրաֆի վերակառուցման մնացած փուլերում կամ ծավալային կենսաբժշկության պատկերների այլ աշխատանքների համար: Իսկ ենթակառուցվածքը կիրառություն ունի խոր ուսուցման ցանկացած ոլորտում:

Աշխատանքի հիմնական նպատակը և դիտարկված խնդիրները: Աշխատանքի հիմնական նպատակն է՝ հետազոտել մեքենայական ուսուցման համակարգերի լայնածավալ տվյալների մշակումը կենսաբժշկական մեծ պատկերների վերադրման խնդրի շուրջը: Աշխատանքում ուսումնասիրվել են մեծածավալ մեքենայական ուսուցման կիրառման խնդիրները, այդ թվում՝ մեծաքանակ տվյալների պահպանումն ու տեղաբաշխումը, զուգահեռացված մոդելների ուսուցումն ու խոր մոդելների հաշվարկումը CPU-ի և GPU-ի կիրառմամբ, ինչպես նաև նրանց արդյունավետության համեմատական վերլուծությունը: Աշխատանքում դիտարկվել է նաև թույլ վերահսկելի եղանակով մոդելների օպտիմիզացիան կենսաբժշկության մեծ նկարների վերադրման խնդրի համար:

Հետազոտության մեթոդները: Հետազոտությունները կատարվել են՝ օգտագործելով թվային մեթոդներով օպտիմիզացիաներ, այդ թվում՝ ասինխրոն և սինխրոն գրադիենտ

[5]Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." In International Conference on Medical image computing and computer-assisted intervention, pp. 234-241. Springer, Cham, 2015.

[6]Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51.1 (2008): 107-113.

[7]Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., Stoica, I. (2010). Spark: Cluster computing with working sets. HotCloud, 10(10-10), 95.

անկում: Բոլոր հիմնական բաշխված համակարգերի փորձերը կատարվել են ամպային լուծումներում

Գիտական նորույթը:

1. Մշակվել է լայնամաշտաբ հաշվարկների համակարգ, որը ընդհանրացվում է տվյալների պահպաման, ցանցերի նախապատրաստման, կոնտեյներների տեղաբաշխման խոր ուսուցման խնդիրների լուծման համար:
2. Մշակվել է խոր ուսուցման գրադարան, որը 3-4 անգամ գերազանցում է Intel Xeon պրոցեսորների հաշվարկային արագությունը, այդ թվում՝ հայտնի PyTorch-ն և Tensorflow-ն ծավալային պատկերների համար:
3. Հետազոտվել և մշակվել է նոր թույլ ուսուցման մեթոդ, որի արդյունքում կենսաբժշկական պատկերների վերադրման սխալանքը 2-7 անգամ նվազել է (հաջորդական նկարների դեպքում՝ 0,11%-ից 0,05% և մոտ նկարների դեպքում՝ 0,47%-ից 0,06%):
4. Առաջարկվել է նոր նեյրոնային ցանցի շերտ՝ նորմալացված փաթույթային շերտ: Այն օգտակար է տարբեր պայծառության պատկերների համընկումը հաշվարկելիս:

Ստացված արդյունքների գործնական կիրառությունը: Միավորելով խոր վերադրման մեթոդը, խոր ուսուցման գրադարանը և ենթակառուցվածքի համակարգը՝ հնարավոր է կիրառել կենսաբժշկության ոլորտում ծավալային մեծ պատկերների վերադրման և տեսողության այլ խնդիրների լուծման նպատակով, մասնավորապես՝ նեյրոգիտության կոնեկտոմիկսի անատոմիական գրաֆի վերակառուցման համար:

Պաշտպանության ներկայացվող հիմնական դրույթները: Պաշտպանության են ներկայացվում հետևյալ հիմնական դրույթները.

1. Մեքենայական ուսուցման ենթակառուցվածքային համակարգ՝ միավորելով տվյալների պահպանումը, վիրտուալ ցանցերի ստեղծումը, բաշխված մեքենաների կառավարումը ամպային լուծումներում:
2. Պետաբայթ մասշտաբի հասնող մեծ զանգվածների և տվյալների պահպանման տեսակ, որը թույլ է տալիս բաշխված նեյրոնային ցանցերի օպտիմիզացիան կենտրոնական տվյալների պահեստից:
3. Ծավալային պատկերների խոր մոդելների մշակման գրադարան մասնագիտացված Intel Xeon պրոցեսորի վրա:
4. Թույլ վերահսկելի մետրիկ/հեռավորության ուսուցման մեթոդ մեծ նկարների վերադրման համար:
5. Փաթույթային նեյրոնային ցանցերի նոր տեսակ՝ նորմալացված փաթույթային ցանցեր:

Ստացված արդյունքների գրաքննությունը և փորձարկումը: Ստացված արդյունքները զեկուցվել են ՀՀ ԳԱԱ ԻԱՊԻ և Հայ-ռուսական համալսարանի ընդհանուր սեմինարներում, Պրինստոնի համալսարանում (ԱՄՆ), ինչպես նաև ներքոհիշյալ

գիտաժողովներում. IARPA MICrONS Phase 2 Meeting July 24-25, 2017; General Examination, Princeton University 2017 October, 2017; Oral Presentation, Computer Vision Conference (CVC) 2019 April 25-26 2019; 12th International Conference on Computer Science and Information Technologies, Yerevan, Armenia, September 23-27, 2019:

Հրապարակումները: Աշխատանքի հիմնական արդյունքները տպագրված են 6 գիտական հոդվածներում, որոնցից երկուսը ընդգրկված են Scopus-ի ցանկում:

Աշխատանքի ծավալը և կառուցվածքը: Աշխատանքի ծավալը կազմում է 107 էջ: Աշխատանքը բաղկացած է ներածությունից, երեք գլուխներից և եզրակացությունից: Աշխատանքը ներառում է 33 նկար և 5 աղյուսակ:

Աշխատանքի պարունակությունը

Մեծ տվյալների հաշվարկներ: Առաջին գլխի 1.1 պարագրաֆում ամփոփվում է մեծ տվյալների հաշվարկների ոլորտը, որը ներառում է պահպանման եղանակները, ամպային լուծումները, զուգահեռացված համակարգերը և մեքենայական ուսուցման գրադարանները:

Վերջին տասնամյակներին տեղեկատվական տեխնոլոգիաների զարգացման արդյունքում, այդ թվում՝ համացանցի գործունեությամբ և բջջային սարքերի համատարած օգտագործմամբ, հավաքվել են մեծ քանակի տվյալներ: Դրանք ներառում են օգտատերերի տվյալների հավաքագրումը, պատկերների բեռնումը սոցիալական ցանցեր, տեսանյութերի հարթակների տվյալների ստեղծումը, ՄՌՏ-ի և ՖՄՌՏ-ի գլխուղեղի պատկերների պահպանումը: Մեծ քանակի տվյալների հետ աշխատանքը կոչվում է մեծ տվյալների հաշվարկումներ[8]: Այն սովորաբար պահանջում է մեծ հաշվարկային ենթակառուցվածք: Աշխատանքի գործողությունները ներառում են տվյալների պահպանում, տվյալների բաշխում, նախամշակում, մոդելավորում և մոդելների տեղակայում:

Տվյալները լինում են երկու տեսակի՝ կառուցվածքային և անկառուցվածքային: Կառուցվածքային տվյալները ունեն հստակ տվյալների տիպեր և սովորաբար պահպանվում են SQL կամ NoSQL տվյալների բազաներում: Անկառուցվածքային տվյալները սովորաբար համարվում են տեքստեր, նկարները, տեսանյութերը և աուդիո ձայնագրությունները: Մեծ տվյալների պահպանման համար հարկավոր է ունենալ բաշխված ֆայլային համակարգ, ինչպես օրինակ՝ ցանցային ֆայլային համակարգը (NFS), հադուպ ֆայլային համակարգը (HDFS) կամ առաձգական բաշխված տվյալների շտեմարանը (RDD): Ամպային լուծումներում հասանելի են նաև օբյեկտային պահպանման եղանակը, օրինակ՝ AWS S3-ը:

Ամպային լուծումները հնարավորություն են տալիս հավելվածի ծրագրավորման ինտերֆեյսի (API) շնորհիվ օգտվել վիրտուալ համակարգչային ռեսուրսների ծառայություններից, օրինակ՝ նշել համակարգչի տեսակը (քանի պրոցեսոր կամ ինչքան հիշողություն) և ժամանակահատվածը: Ընկերությունները կարիք չունեն նախապես գնելու թանկարժեք սերվերներ և կարող են օգտվել ամպային լուծումներից

[8]«Մեծ» բառի կիրառումը հարաբերական է տվյալ ժամանակաշրջանի մեկ համակարգչի հնարավորությունների նկատմամբ: 2019թ. դրությամբ 10 ՏԲ-ից մինչև 1 ՊԲ ավելի հասնող տվյալները համարվում են մեծ տվյալներ:

ըստ պահանջի: Կան հավելյալ ծառայություններ, որոնք հեշտացնում են աշխատանքը ամպային տարածքում: Օրինակ՝ Kubernetes համակարգը թույլ է տալիս կոնտեյներների տեղաբաշխումը հազարավոր մեքենաների վրա:

MapReduce պարադիգմի վրա հիմնված զուգահեռացված տվյալների համակարգերը, ինչպես օրինակ՝ Hadoop-ը և Spark-ը, կարող են գործածվել բաշխված վիրտուալ համակարգերում: Տվյալագիտության ոլորտում կան տարբեր գրադարաններ, որոնք թույլ են տալիս տվյալները հստակ ներկայացնել աղյուսակների կամ տենսորային զանգվածների տեսքով: Գոյություն ունեն մոդելավորման գրադարաններ՝ նախատեսված տարբեր կիրառումների համար, ինչպիսիք են, օրինակ, SciKit, XGBoost, TensorFlow, PyTorch և այլն:

Մեքենայական ուսուցման հիմունքներ: 1.2 պարագրաֆը նվիրված է մեքենայական ուսուցման հիմունքներին: Այն ներառում է վիճակագրական ուսուցումը, օպտիմիզացիան և խոր ուսուցումը: Հայտնի են մեքենայական ուսուցման մի քանի խնդիրներ (վերահսկելի, անվերահսկելի, առցանց և ամրացման խնդիրներ): Վերահսկելի ուսուցումը ձևակերպվում է այսպես՝ տրված են մուտք-ելք փոփոխականներ՝ $S = (x, y)_i^n \sim D^n$, որտեղ $x \in X$ (մուտքի տարածություն), $y \in Y$ (ելքի տարածություն) և տվյալների բաշխում D $X \times Y$-ում, անհրաժեշտ է գտնել $f : X \to Y$, որը նվազեցնում է սպասված ստուգման սխալմունքը:

$$f_\star = argmin_f E_{(x,y)\sim D}[L(f(x); y)] \tag{1}$$

L-ը սխալանքի չափն է: Օպտիմիզացիայի խնդիրն է գտնել $f \in F$, որը հասնում է մինիմալ ուսուցման սխալմունքի: Նկատենք, որ սա NP-բարդ խնդիր է, սակայն կարելի է ընտրել այնպիսի F, որի դեպքում հնարավոր են արդյունավետ գործնական ալգորիթմներ:

Վերցնենք $f_\theta \in F$ ածանցելի պարամետրացում F-ում, որտեղ $\theta \in R^{|\theta|}$ և $|\theta|$ պարամետրերի քանակն է: Սահուն փոփոխելով θ-ն կարելի է ներկայացնել ցանկացած ֆունկցիա F-ում: Այսպիսով խնդիրը ձևակերպվում է՝ գտնել θ-ն սահուն տարածության մեջ: Քանզի ֆունկցիան ածանցելի է, կարելի է օգտագործել գրադիենտ անկումը:

$$\theta_{t+1} \leftarrow \theta_t - \bigtriangledown f(\theta_t) * \epsilon \tag{2}$$

Եթե հաշվարկները կատարվում են բաշխված համակարգում, ապա պարամետրերի համաժամացումը կատարվում է ասինխրոն կամ սինխրոն գրադիենտ անկման մեթոդների կիրառմամբ: Նեյրոնային ցանցերը համարվում են լավագույն F-ի ընտրություններից մեկը խնդիրներում, ինչպիսիք են, օրինակ, համակարգչային տեսողության, ձայնագրությունների, բնական լեզվի և այլ ոլորտի խնդիրները: Նեյրոնային ցանցերը բաղկացած են շերտերից: Ամեն մի շերտ կատարում է գծային վերափոխում ($Wx + b$) և կիրառում տարրական ոչ գծային ֆունկցիա:

$$\begin{aligned} h_1 &= f(W_1 x + b_1) \\ h_2 &= f(W_2 h_1 + b_2) \\ &\dots \\ y &= f(W_n h_{(n-1)} + b_n) \end{aligned} \tag{3}$$

Այստեղ θ=$(W_i, b_i)_n$, որտեղ n-ը շերտերի քանակն է: Խոր ուսանումը օգտագործում է հետտարածման ալգորիթմը, որպեսզի հաշվվի ամեն մի փոփոխականի վրա ընկնող

մասնակի աճանցյալը: Ապա կիրառվում է գրադիենտ անկման ալգորիթմը: Հայտնի են տարբեր նեյրոնային ցանցերի կառուցվածքներ. Օր.՝ փաթույթային ցանցեր, ռեկուրենտ ցանցեր և տրանսֆորմատորներ:

Կիրառումը կենսաբժշկության ոլորտում: Առաջին գլխի 1.3 պարագրաֆում ներկայացնում ենք արդի կենսաբժշկության պատկերների խնդիրները և կենտրոնանում նկարների վերադրման մեթոդների վրա առանց ընդհանրության կորստի: Համակարգչային տեսողության կարևոր կիրառություններից է կենսաբժշկության պատկերների մշակումը: Այն ներառում է ՀՏ (հաշվարկված տոմոգրաֆիա), ռենտգեն, ՄՌՏ (մագնիսական ռեզոնանսային տոմոգրաֆիա), ՖՄՌՏ (ֆունկցիոնալ մագնիսական ռեզոնանսային տոմոգրաֆիա), ԷՄ (էլեկտրոնային մանրադիտակ) մշակման խնդիրները: Խոր ուսուցումն առաջին անգամ գերազանցեց բժիշկներին՝ ճշգրտելով կրծքավանդակի քաղցկեղի հայտնաբերման բժիշկների դասակարգման սխալանքները[9]:

Նեյրոգիտությունը մեկ այլ կարևորագույն կիրառություններից է: Կոնեկտոմիկսի խնդիրն է՝ գլխուղեղի մանրադիտակի նկարներից վերակառուցել նեյրոնների կապերի գրաֆը: Այն հետագայում թույլ կտա հետազոտել բնական ուղեղի ուսուցման ալգորիթմերը և հայտնաբերել Ալցհայմերի կամ Պարկինսոնի հիվանդության պատճառները: Առաջին ամբողջական կոնեկտոմը հայտնի է նեմատոդ C. elegans [10], որը ունի 300 նեյրոն և 7000 սինապս: Ձեռքով վերամշակմամբ կապերի վերականգնումը տևեց մեկ տասնամյակ 20-րդ դարում: Օգտագործելով խոր ուսուցման մեթոդներ բաշխված մեքենաների վրա՝ հնարավոր է վերակառուցել առավել մեծ կենդանիների ամբողջական նեյրոնների կապեր, օրինակ՝ ճանճի (100 ՏԲ), մկան (1000 ՊԲ կամ 1 ԷԲ) կամ մարդու (1000 ԷԲ), նույն կամ ավելի լավ որակի:

Նեյրոնների անատոմիական վերակառուցման առաջին խնդիրը վերաբերում է մեծաքանակ (2000 հատ 100000x100000 պիկսելային չափի) պատկերների վերադրմանը: Սաաֆիլդը[11] ներկայացրել է էներգետիկ զսպանակների մեթոդով մոդելավորման էլաստիկ փոխակերպումը: Սակայն կան ձևափոխություններ, որոնք էլաստիկ չեն: Խոր ուսուցմամբ կարելի է սովորելով բարելավել կամ ամբողջովին լուծել վերադրման խնդիրը: Քանի որ գոյություն չունեն ճշմարիտ պիտակներ, խնդիրը պետք է լուծել անվերահսկելի ուսուցմամբ: Վերադրումից հետո կատարվում է նեյրոնների հատվածավորում (segmentation) և սինապսների հայտնաբերում:

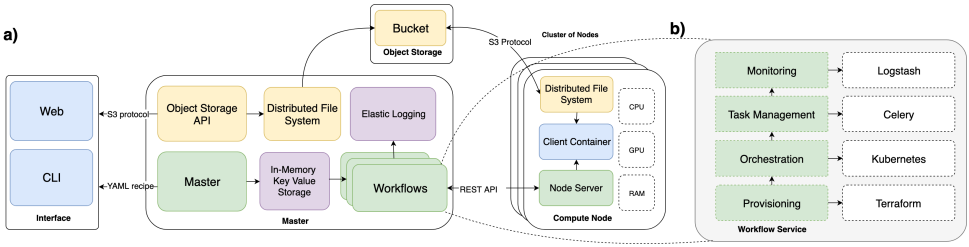
խոր ուսուցման խնդիրների բաշխված ամպային հաշվարկում

Մեծ տվյալների արդյունավետ մշակման համար հաշվարկային և պահպանման միավորները պետք է կիրառվեն որպես մեկ սուբյեկտ: Բաշխված հաշվարկումը պատասխանատու է առաջադրանքների պլանավորման և տեղակայման համար, այդ թվում՝ տվյալների, միջավայրի, ծրագրի և ալգորիթմների, մեկ կամ մի քանի վիրտուալ մասնավոր ցանցերում: Ամպային լուծումները, ինչպես օրինակ՝ AWS, GCP կամ Azure, թույլ են տալիս, օգտագործելով ծրագրավորային ինտերֆեյս, ըստ պահանջի բարձրացնել և կառավարել հաշվարկային միավորներ: Այն տալիս է մուտք մանրակրկիտ

[9]Rajpurkar, Pranav, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning." arXiv preprint arXiv:1711.05225 (2017).

[10]White, John G., Eileen Southgate, J. Nichol Thomson, and Sydney Brenner. "The structure of the nervous system of the nematode Caenorhabditis elegans." Philos Trans R Soc Lond B Biol Sci 314, no. 1165 (1986): 1-340.

[11]Saalfeld, Stephan, Richard Fetter, Albert Cardona, and Pavel Tomancak. "Elastic volume reconstruction from series of ultra-thin microscopy sections." Nature methods 9, no. 7 (2012): 717.



Նկ. 2.1: Համակարգի ճարտարապետությունը. a) Ինտերֆեյսից վերբեռնվում են ուսուցման տվյալները, ծրագիրը և YAML բաղադրատոմսը Master հաշվարկման միավորին: Ֆայլերը տրոհվում են մասերի և պահպանվում օբյեկտային հիշողությունում: Բաղադրատոմսը ներկայացվում է հաշվարկային աշխատանքահոսքի տեսքով և պահպանվում է բանալի-արժեք հիշողության մեջ: Յուրաքանչյուր աշխատանքային հոսքի համար ստեղծվում է ամպային կլաստեր: Յուրաքանչյուր հաշվարկման միավոր ունի ունկնդիր սերվեր, որը կառավարում է նույն միավորի աշխատանքը և կատարում հրամանները: b) Աշխատանքահոսքը ունի չորս փուլ՝ ենթակառուցվածքի ստեղծում, միավորների նվագախումբ, առաջադրանքների կատարում և միավորների մոնիտորինգ:

ծառայություններին: Օրինակ՝ թույլ է տալիս ստեղծել հիշողության դույլեր կամ տեղակայել վիրտուալ ցանցեր: Սակայն բաշխված համակարգը նաև պետք է կառավարի օպերացիոն համակարգը և տվյալների պահպանման շերտը:

Աշխատանքում ներկայացնում ենք խոր ուսուցման բաշխված հաշվարկների համակարգ, որը իր վրա է վերցնում ռեսուրսների կառավարումը, առաջադրանքների հերթագրումը, տվյալների կառավարման շերտը և թույլ է տալիս արդյունավետ ուսանել, կիրառել և տեղակայել մեքենայական ուսուցման մոդելներ ամպերում: Այն թույլ է տալիս մշակել ժամերի ընթացքում տեռաբայթի հասնող ինֆորմացիա և հեշտությամբ հասնել պետաբայթի ծավալով տվյալների մշակման 30 ՊՖԼՕՊ հզորությամբ:

2.2 պարագրաֆում ներկայացվում են առաջարկված համակարգի հաշվարկային մոդելը և աշխատանքային ինտերֆեյսը: Աշխատանքային հոսքը (*Workflow*) ուղղորդված ցիկլ չպարունակող գրաֆ է՝ բաղկացած փորձարկման գագաթներից (*Experiment*) և նրանց կախվածությունը ներկայացնող կապերից: Մեկ փորձը պարունակում է մի քանի առաջադրանք: Մեկ փորձի յուրաքանչյուր խնդիր կատարում է միևնույն հրամանը տարբեր արգումենտներով: Առավել արդյունավետ փոփոխականների բաշխման համար արգումենտները կարող են ներկայացվել ձևանմուշի տեսքով: Յուրաքանչյուր փորձ ունի իրեն կցված կոնտեյներ, որը տեղակայված է բոլոր հաշվարկման միավորներում: Մեկ առաջադրանքը սովորաբար կատարում է մեկ պրոցես, յուրաքանչյուր առաջադրանք՝ մեկ *Node*, որը հաշվարկման միավորն է: Մեկ *Node*-ը կարող է հաջորդաբար կատարել մի քանի առաջադրանք:

Workflow-ն հայտարարվում է ծրագիր-որպես-ենթակառուցվածքի ինտերֆեյսով որպես YAML բաղադրատոմս: Այն թարգմանվում է ուղղորդված էքսպերիմենտների ուղղորդված ցիկլ չպարունակող գրաֆի: Ինտերֆեյսը թույլ է տալիս նշել միջավայրը, համակարգչի նկարագրությունը, հաշվարկային միավորների քանակը, պարամետրերը և ձևանմուշային հրամանները:

Օգտատերը հայտարարում է պարամետրերի ցանկը, որը հաշվարկի ժամանակ տեղակայվում է հրամանի մեջ: Պարամետրերը կարող են վերցվել դիսկրետ կամ շարունակական միջակայքերից: Ալգորիթմը գեներացնում է դիսկրետ պարամետրերի կարտեզյան արտադրյալը, ապա արտադրյալից վերցնում n նմուշ ամենաքիչ

կրկնողությամբ: Այնուհետև գեներացնում է պատահական n թիվ անընդհատ միջակայքից և համակցում դիսկրետ նմուշներին: Այս մեթոդը թույլ է տալիս կատարել և՛ հիպերպարամետրերի որոնում, և՛ մեծ պարամետրավորված հաշվարկներ:

2.3 պարագրաֆում ներկայացված են բաշխված համակարգի մանրամասները:

Ֆայլային համակարգ: Ֆայլային համակարգը տրոհում ենք և պահպանում ամպային հիշողության դույլերում (օրինակ՝ AWS S3), որը նույնն է օբյեկտային հիշողությունում: Բաշխված ֆայլային համակարգը փաթեթավորում է POSIX ինտերֆեյսը և ապահովում կտորների տրոհումը, պահոցային հիշողությունը և սինխրոնիզացիայի մեխանիզմը:

Երբ ծրագիրը հարցում է կատարում ինչ-որ ֆայլի համար, ինտեգրման շերտը ստուգում է կտորների տեղակայումը և ներբեռնում այն կտորները, որոնք տեղում չեն: Ծրագրի կոնտեքստում հեռակա պահպանվող օբյեկտային կտորները ներկայացվում են որպես տեղական ֆայլեր: Ցանկացած խոր ուսուցման հավելված առանց փոփոխության կարող է օգտագործել ֆայլային համակարգը մասշտաբային հաշվարկներ կատարելիս: Ցանցային ֆայլային համակարգը (NFS) կարող է ապահովել մեծ ներբեռնման ու վերբեռնման հնարավորություն և ունակ է պահպանելու ֆայլային համակարգի (HDFS) ցանկացած ֆայլ:

Խոր ուսուցման միջավայրերը թույլ են տալիս ասինխրոն տվյալների տեղափոխումը տեղական ֆայլային համակարգից դեպի GPU-ի հիշողություն: Եթե համակցվեն հեռակա տվյալների պահպանումն ու ասինխրոն տեղափոխումը, ապա հնարավոր է ստանալ ուսուցման բարձր արագություն՝ տվյալների տեղական պահպանմանը համարժեք:

Նախապատրաստում: Քանի որ համակարգը թույլ է տալիս ցանկացած գրադարանի օգտագործում, ապա ամբողջ միջավայրը պետք է տեղափոխվի հաշվարկային մեքենայի վրա: Հաշվարկային մեքենան պետք է պարունակի Docker և Nvidia CUDA ծրագրային ապահովում: Վիրտուալ մեքենայի պատկերը կարող է հիմնվել ցանկացած UNIX օպերացիոն համակարգի վրա, այդ թվում՝ CoreOS, Ubuntu կամ CentOS: Վիրտուալ մեքենայի պատկերը ստեղծվում է մեկ անգամ:

Նվագախմբավորում (Orchestration): Համակարգի նվագախմբային պրոցեսը ընդգրկում է կոնտեյներների տեղափոխումը ՎՄ-ներ և նրանց աշխատանքը: Համակարգը նման է Kubernetes-ին իր պարտականություններով, սակայն տարբերվում է՝ պահպանելով կոնտեյների վիճակը: Նաև պարունակում է տեղեկամատյան՝ հավաքելով պրոցեսի ելքը, օպերացիոն համակարգի վիճակը և CPU/GPU օգտագործումը:

Ցանցերի կառավարում: Ցանկացած առաջադրանքի համար համակարգը կարգաբերում է համացանցի դարպասով (Internet Gateway) վիրտուալ մասնավոր ցանց: Այն համակարգում է հաշվարկային միավորների ներքին կապը այն խնդիրներում, որոնք ներառում են միավորների սինխրոնիզացիա, օրինակ՝ բաշխված սնուցման ժամանակ: Այլընտրանքային լուծում է բաշխված ֆայլային համակարգի օգտագործումը որպես սերվերի առանց ցանցային կարգաբերման պարամետր:

Համակարգը: Նկ. 2.1-ից հետևում է, որ առաջարկված համակարգի ճարտարապետությունը բաղկացած է ինտերֆեյսից, *Master*-ից և հանգույցներից (Node): *Master*-ը պատասխանատու է աշխատանքահոսքի բաղադրատոմսը վերլուծելու

և տրոհելու էքսպերիմենտ և առաջադրանք օբյեկտների: Օբյեկտները պահպանվում են in-memory key-value հիշողության պահոցում՝ օգտագործելով Redis համակարգը: Նշենք նաև, որ օբյեկտները կրկնօրինակվում և պահպանվում են ոչ ռելացիոն համակարգում: *Master*-ը ստեղծում է հարակից կոնտեյներ յուրաքանչյուր նոր աշխատահոսքի ծառայությունը սկսելու և առաջադրանքները ժամանակադրելու համար: Նվագախմբային պրոցեսի ժամանակ բոլոր միավորները աշխատեցնում են ունկնդիր սերվերներ և կատարում *Master*-ի հրամանները:

Հանդուրժողականություն: Հաշվարկային միավորների ռեսուրսային ծախսը 3 անգամ օպտիմիզացնելու համար ամպային լուծումները տրամադրում են տատանվող մեքենաներ: Նրանք կարող են կանգ առնել ցանկացած ժամանակ՝ կախված հայտարարված ժամանակավոր գնից: Երկարատև պրոցեսների կատարման համար սխալմունքային հավանական համակարգում հարկավոր է հավելյալ հաշվարկային տրամաբանություն: Միավորելով բաշխված ֆայլային համակարգը և առաջադրանքների պլանավորման համակարգերը՝ հնարավոր է նվազագույն ընդհատումով օգտագործել տատանվող մեքենաները:

Քննարկում: *Hyper* համակարգը թույլ է տալիս միավորել բազմակի ամպային պլատֆորմներ և տեղական հաշվարկային կլաստերներ առանց DevOps ծրագրավորողների: Այն խնայում է ռեսուրսների ծախսը և տրամադրում համակարգի մասին վիճակագրություն: *Hyper*-ը օգտագործվել է հարյուրավոր գիտնականների կողմից: Մի քանի ընկերություն օգտագործում են իրենց նեյրոնային ցանցերի ուսուցման համակարգերում:

Ի տարբերություն Map-Reduce-ի հիմնած համակարգերի, ինչպես նաև Hadoop-ի կամ Spark-ի, և ուղղորդված գրաֆի հաշվարկման գրադարանները, ինչպես օրինակ՝ Dask կամ Ray[12] համակարգերը, *Hyper* բաշխված համակարգը արդյունավետ է խոր ուսուցման հաշվարկների համար:

Համակարգի թեստավորում: Երրորդ գլխի 3.1 պարագրաֆը ներառում է համակարգի ստուգումը 4 հիմնական խոր ուսուցման խնդիրներում: *Առաջին խնդրում* կատարում ենք 100 միլիոն փաստաթղթի վերամշակում: Տեքստի վերամշակումը ներառում է տեքստը մաքրելը, նախադասությունները տրոհելը և նշանավորելը: Տվյալները հասնում են 10 ՏԲ ծավալի: YAML ինտերֆեյսով նշում ենք 110 մեքենա, յուրաքանչյուրը՝ 96 պրոցեսորով: Յուրաքանչյուր առաջադրանք վերցնում է 100000 փաստաթուղթ և վերամշակում tfrecord ֆայլերի տեսքով:

Երկրորդ խնդրում կատարում ենք բաշխված ուսուցում՝ օգտագործելով սինխրոն գրադիենտ անկման ալգորիթմը: Պատկերների ճանաչման խնդրում վերբեռնում ենք COCO[13] տեսադարանը և օգտագործում YOLO[14] խոր ուսուցման մոդելը: Բաղադրատոմսի մեկ տողի տարբերությամբ կարող ենք օգտագործել 8 Nvidia V100 և տատանման

[12]Moritz, Philipp, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol et al. "Ray: A distributed framework for emerging AI applications." In 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), pp. 561-577. 2018.

[13]Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In European conference on computer vision, pp. 740-755. Springer, Cham, 2014.

[14]Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. 2016.

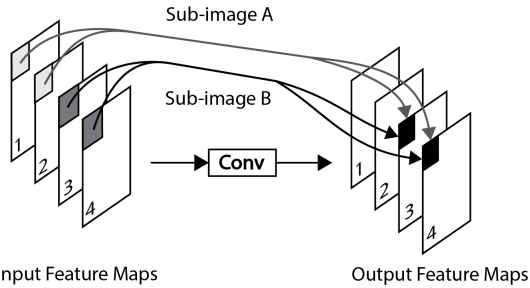
մեքենաներ, մեկ Nvidia K80-ի համեմատ ստանալով 50 անգամ ավելի արագություն և 6 անգամ արդյունավետություն:

Երրորդ խնդրում կատարում ենք գրադիենտ խթանող ծառերի հիպերպարամետրերի որոնում: Յուրաքանչյուր ուսուցում մեկ հաշվարկման մեքենայի վրա տևում է 10 րոպե: Կան 4096 տարբեր պարամետրեր: Հաջորդական 28 օրվա հաշվարկման փոխարեն օգտագործելով մեր համակարգը կարելի է միևնույն ժամանակ ստուգել բոլոր պարամետրերը 10 րոպեում գծային հորիզոնական մեծացմամբ և առանց ծրագրի փոփոխման:

Չորրորդ խնդրում կատարում ենք 450 հազար պատկերներում օբյեկտների հայտնաբերում և դասակարգում՝ օգտագործելով երկրորդ խնդրի ստացած մոդելը: Օգտագործում ենք միաժամանակ 300 GPU՝ հասնելով 30 ՊՖԼՕՊ մաքսիմալ հզորության:

Կենտրոնացած պրոցեսորի խոր ուսուցման միջավայր

Երրորդ գլխի 3.2 պարագրաֆում ներկայացնում ենք խոր ուսուցման միջավայր Intel Xeon պրոցեսորների համար, որը գերազանցում է Tensorflow և Pytorch համակարգերին համարժեք հաշվարկման դեպքում և 1,5 անգամ խնայում է ծախսը GPU-ների համեմատ: Փաթույթային շերտը հաշվելու համար օգտագործում ենք ZnnPhi ալգորիթմը[15]:



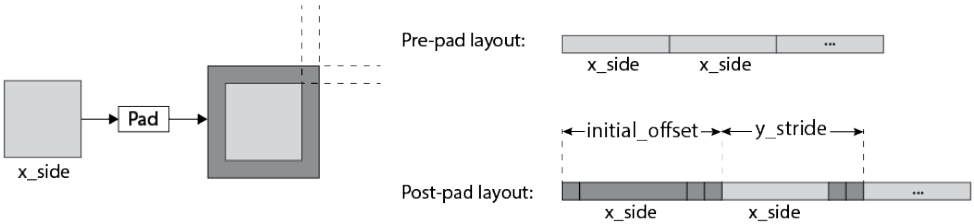
Նկ. 3.1: Երկու ZnnPhi նկարի պրիմիտիվների կիրառությունը դեպի ելքի սև վանդակների ($SIMD\_width = 2$): Յուրաքանչյուր կիրառություն հաշվում է $SIMD\_width$ հատկությունների պատկերների ներդրումը ելքի սև վանդակներում: A և B կիրառելուց հետո ելքի վանդակները կպահպանեն իրենց արժեքը:

Գումարային միաձուլում: Խոր ուսուցման մոդելները հաճախ պարունակում են մնացորդային կապեր: Մնացորդային կապերը սովորաբար հաջորդիվ փաթույթային շերտեր են, որտեղ առաջին շերտը ունի նաև հավելյալ կապ վերջին շերտի հետ, տեղային գումարման օպերատորով գումարային միաձուլումը համախմբում է տեղային գումարման օպերացիան փաթույթային շերտի հետ:

Գծային փոխանցման միաձուլում: Խմբային նորմալիզացիայի շերտը թույլ է տվել բարելավել ուսուցման արդյունավետությունը և ընդհանրացման սխալը[16]: Փոփոխելով

[15]Zlateski, Aleksandar, Kisuk Lee, and H. Sebastian Seung. "ZNN–A Fast and Scalable Algorithm for Training 3D Convolutional Networks on Multi-core and Many-Core Shared Memory Machines." In 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 801-811. IEEE, 2016.

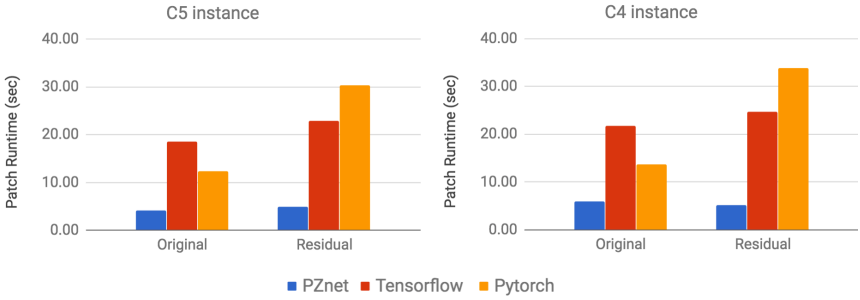
[16]Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).



Նկ. 3.2: Ձախում՝ ավելացման շերտի օպերատոր: Աջում՝ հիշողության հարթություն մինչ շերտի ավելացումը և շերտի ավելացումից հետո: Ավելացված եզքը կարող է դիտվել որպես մուտքի պատկերի ներկայացում. $\text{initial}_offset = x_side + 3y_stride = x_side + 2$

:

փաթույթային շերտի միջուկի պարամետրերը՝ կարելի է միաձուլել գծային տեղափոխման օպերացիան: Այսպիսով, խմբային նորմալիզացիան կարող է ամբողջովին վերացվել՝ պահպանելով հաշվարկային ճշգրտությունը մոդելի տեղակայման ժամանակ:



Նկ. 3.3: Խոր ուսուցման մշակված PZNet մեթոդի համեմատությունը հայտնի Tensorflow և PyTorch համակարգերի հետ (որքան քիչ, այնքան լավ):

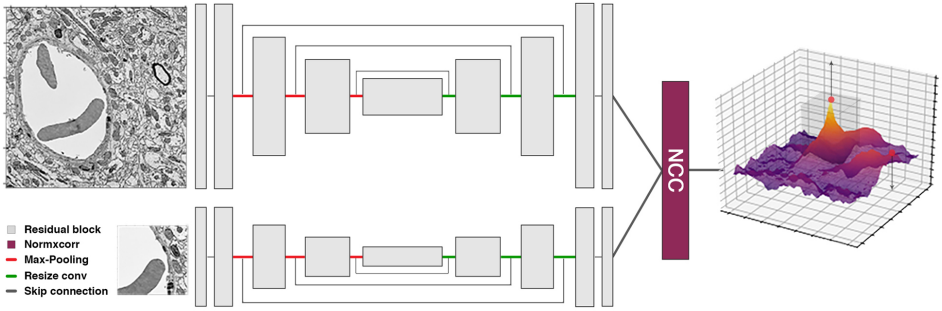
Ակտիվացման ֆունկցիայի միաձուլումը: Փաթույթային շերտից հետո բոլոր ակտիվացիոն ֆունկցիաները կարելի կիրառել ռեգիստրներում: Այն խնայում է տվյալների տեղափոխման ժամանակը L1, L2 և L3 պահոցներ:

Նաև կատարում ենք հիշողության դասավորության օպտիմիզացիա փաթույթային շերտերի համար և, օգտագործելով SIMD հրահանգները, ստանում ենք առավել արդյունավետ համակարգ, քան հայտնի խոր ուսուցման MKL-ի վրա հիմնված տարբերակները կենտրոնական պրոցեսորների վրա: Այս արդյունավետությունը նաև թույլ է տալիս 1,5 անգամ խնայել ծախսը GPU-ների նկատմամբ:

Պատկերների խոր վերադրում

Երրորդ գլխի 3.3 պարագրաֆում քննարկում ենք պատկերների վերադրման խնդիրը և առաջարկում ենք թույլ վերահսկելի մետրիկայի ուսուցման մեթոդ:

Տրված է պատկերի մաս՝ $T \in R^t$ և ավելի մեծ պատկեր՝ $F = \{F_i : F_i \in R^t\}_n$ (բոլոր հնարավոր երկու պատկերների համընկումների քանակի), նորմալիզացված



Նկ. 3.4: Siamese U-Net ցանցերի ելքը կապվում է NCC շերտի հետ:

փաթույթային շերտը ձևակերպվում է:

$$NCC(F_i, T) = \frac{(F_i - \mu_{F_i})^\top (T - \mu_T)}{\sigma_{F_i}\sigma_T} \tag{4}$$

որտեղ μ-ն միջին և σ-ն դիսպերսիան է:

Նորմալիզացված շերտի ելքը պատկեր է, որտեղ յուրաքանչյուր պիկսել տալիս է T-նկարի կոսինուսային նմանությունը F_i կոորդինատում (բանաձև 4):

Վերցնենք P_1 առավելագույն նորմալիզացված փաթույթային շերտի ելքի գագաթը և երկրորդ առավելագույն գագաթը P_2 սահմանափակ նվազագույն հեռավորության վրա:

$$\begin{aligned} P_1(F, T) &= \max_i NCC(F_i, T) \\ P_2(F, T) &= \max_{i^* \neq j} NCC(F_j, T) \end{aligned} \tag{5}$$

Օպտիմիզացիայի նպատակն է՝ մեծացնել առաջին և երկրորդ գագաթների հեռավորությունը $L(F, T) = P_1(F, T) - P_2(F, T)$:

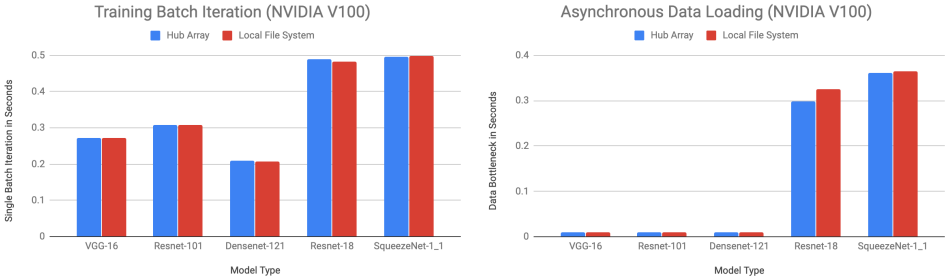
$$\max_{\psi \in \Psi} \quad L(\psi(F), \psi(T)) \tag{6}$$

Պատկերների փոխակերպման մոդելն օգտագործում ենք Siamese նեյրոնային ցանցերի ներկայացման համար (տես՝ նկ. 3.4-ը), որի արդյունքում փոքր պատկերների վերադրման սխալանքը քչանում է 2-7 անգամ (հաջորդական նկարների դեպքում՝ 0,11%-ից 0,05%, և մոտ նկարների դեպքում՝ 0,47%-ից 0,06% տես աղյուսակ 3.1): Քննարկում ենք նաև թույլ վերահսկելի մեթոդով սովորած մոդելի մեկնաբանելի հատկությունների փոխակերպման արդյունքում, օրինակ, միթոքոնդրիաների հայտնաբերման կամ պատկերների արատների մերժման հարցերը:

Երրորդ գլխի 3.4 և 3.5 պարագրաֆներում առաջարկում ենք վերադրման ամբողջական խոր ուսուցման համակարգ, որը ուսանում է անվերահսկելի մեթոդով: Ապա առաջարկում ենք համախմբել բաշխված համակարգը, խոր ուսուցման մեր միջավայրը և վերադրման խոր մոդելը, որպեսզի վերադրվեն դրոզոֆիլիայի գլխուղեղի պատկերները (100ՏԲ):

	Adjacent (հաջորդական)				Across (մոտ)			
Փոքր պատկերի չափս	160px		224px		160px		224px	
Raw	1,778	1.23%	827	0.57%	2,105	2.91%	1,068	1.48%
Bandpass	480	0.33%	160	0.11%	1,504	2.08%	340	0.47%
NCCNet	**261**	**0.18%**	**69**	**0.05%**	**227**	**0.31%**	**45**	**0.06%**

Աղյուսակ 3.1: Ցանցը բարելավում է մնացած հայտնի մեթոդները այս առաջադրանքում: Թեստավորման քանակը բաղկացած է 144500 հարակից և 72306 հաջորդող հանդիպումներից:



Նկ. 3.5: Սովորեցնում ենք տարբեր խոր մոդելներ տեղական (կարմիև) և հեռակա ֆայլային համակարգից: Ցույց ենք տալիս, որ անկախ մոդելից՝ հեռակա և տեղային ֆայլային համակարգերը համարժեք են:

Իրականացման մանրամասներ: Չորրորդ գլխի 4.1 պարագրաֆը ներկայացնում է հիպեր համակարգի բաշխված իրականացման մանրամասները: Այն խոսում է հնարավոր վիճակների և համակարգի կառուցվածքի մասին: Նաև ներառում է սխալների հանդուրժողականության և դասակարգման դեպքերի մանրամասն քննարկում:

Չորրորդ գլխի 4.2 պարագրաֆում ներկայացնում ենք հիպերհամակարգի բաշխված պահպանման մեթոդի կիրառությունները և արդյունքները: Ցույց ենք տալիս նկ. 3.5-ով, որ հնարավոր է հեռակա տվյալների պահպանման տեսքով կատարել մոդելների ուսուցանում: Նաև ներկայացնում ենք մասնակի դեպքը՝ զանգվածների կառավարումը, և խոսում տարբեր կիրառությունների մասին:

Չորրորդ գլխի 4.3 պարագրաֆում ներկայացվում են նոր նորմալիզացված փաթույթային շերտի մանրամասները: Այն ներառում է երկու տեսակի իրականացում, որոնցից մեկը օգտագործում է ֆուրերի ձևափոխում օգտագործելով փաթույթների թեորեմը:

Հիմնական արդյունքներն ու հետևությունները

Աշխատանքում դիտարկվել են բաշխված խոր ուսուցման խնդիրներ և առաջարկվել է համակարգ, որը թույլ է տալիս նախամշակում, բաշխված ուսուցում, հիպերպարամետրերի որոնում և մոդելների տեղակայում մեծ տվյալների աշխատանքի համար: Ապա ներկայացվել է խոր ուսուցման միջավայր կենտրոնական պրոցեսորների

հաշվարկումների համար: Նաև ներկայացվել է նկարների վերադրման խոր մոդել: Համախմբելով երեք մեթոդները՝ կարելի է լուծել կենսաբժշկության մեծ նկարների վերադրման խնդիրը: Համակարգը կարող է նաև օգտագործվել այլ մեծ տվյալների խոր ուսուցման խնդիրներում:

Աշխատանքում ստացվել են հետևյալ արդյունքները:

1. Մեքենայական ուսուցման ենթակառուցվածքային համակարգ, որը հեշտացնում է մեծ տվյալների հետ աշխատանքը ամպային լուծումներում: Հնարավորություն է տալիս միաժամանակ հեշտորեն կառավարելու հարյուրավոր մեքենաներ: Համակարգը ստուգվել է 10000 CPU միջուկի և 300 GPU-ների վրա՝ հասնելով 30 ՊՖԼՕՊ հզորության:
2. Մեծ զանգվածների և տվյալների պահպանման տեսակ, որը թույլ է տալիս բաշխված նեյրոնային ցանցերի օպտիմիզացիան ուղիղ կենտրոնական տվյալների պահեստից: Իր արագությամբ այն համարժեք է տվյալների պահպանման տեղական հիշողությունում, սակայն կարող է ընդլայնվել մինչև պետաբայթի հասնող տվյալների:
3. Խոր ուսուցման գրադարան մասնագիտացված Intel Xeon պրոցեսորի վրա: Գերազանցում է 3-4 անգամ հաշվարկների արագության մեջ հայտնի խոր ուսուցման գրադարանները, այդ թվում՝ TensorFlow-ն և PyTorch-ը ծավալային կենսաբժշկական պատկերների մշակման դեպքում: Այն թույլ է տալիս մինչև 1,5 անգամ ավելի էժան հաշվարկների կատարում GPU-ների համեմատ:
4. Թույլ անվերահսկելի մետրիկ ուսուցման մեթոդ: Այն թույլ է տալիս կատարելագործել մեծ նկարների վերադրումը և նվազեցնել 2-7 անգամ սխալմունքը (հաջորդական նկարների դեպքում՝ 0,11%-ից 0,05% և մոտ նկարների դեպքում՝ 0,47%-ից 0,06%):
5. Նոր նեյրոնային փաթույթային ցանցի տեսակ՝ նորմալիզացված փաթույթային շերտ: Էվկլիդյան հեռավորությունը հաշվարկելու փոխարեն, ինչպես միջուկը, այնպես էլ պատկերը նորմալիզացվում են, արդյունքում հաշվարկվում է կոսինուսի նմանությունը:

Ատենախոսության թեմայի շրջանակներում հրապարակված աշխատանքների ցանկ

1. Buniatyan, D., Macrina, T., Ih, D., Zung, J. and Seung, H.S., 2017. Deep learning improves template matching by normalized cross correlation. arXiv preprint arXiv:1705.08593.
2. Buniatyan, D., Popovych, S., Ih, D., Macrina, T., Zung, J. and Seung, H.S., 2019, April. Weakly Supervised Deep Metric Learning for Template Matching. In Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1, p. 39-58. Springer.
3. Popovych, S., Buniatyan, D., Zlateski, A., Li, K. and Seung, H.S., 2019, April. PZnet: Efficient 3D ConvNet Inference on Manycore CPUs. In Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1, p. 369-383. Springer.

4. Mitchell, E., Keselj, S., Popovych, S., Buniatyan, D. and Seung, H.S., 2019. Siamese encoding and alignment by multiscale learning with self-supervision. arXiv preprint arXiv:1904.02643.
5. Buniatyan, D., 2019, September. Hyper: Distributed Cloud Processing for Large-Scale Deep Learning Tasks. Computer Science and Information Technologies (CSIT) (pp. 49-52).
6. Buniatyan, D. 2019, November. "Hyper: Distributed Cloud Processing for Large-Scale Deep Learning Tasks." In 2019 CSIT, pp. 27-32. IEEE, 2019.

Abstract

To train and deploy deep learning models in real-world applications requires processing large amounts of data. This is a challenging task when the amount of data grows to a hundred terabyte, or even petabyte scale. Applications include self-driving cars, aerial image analysis, error detection for industrial use and biomedical image processing. Biomedical image processing deals with large amounts of data where labels are often not available. Images can be distorted, damaged and contain defects.

We introduce a hybrid distributed cloud framework with a unified view of multiple clouds and on-premise infrastructure for processing tasks using both CPU and GPU compute instances at scale. The system implements a distributed file system and failure-tolerant task processing scheduler, independent of the language or Deep Learning framework used. This allows us to utilize unstable cheap resources on the cloud to significantly reduce costs for biomedical image processing.

In addition to optimization across nodes, we introduce a CPU-only inference engine for processing images in cost-efficient manner on a single node. The inference engine extends ZnnPhi[17] by introducing JIT compiler. During compilation, the framework fuses layers, optimizes the memory layout and leverages SIMD instructions for fast processing.

We introduce a weakly supervised metric learning approach to find templates across sections of the biological specimen which significantly enhances the accuracy of the alignment. Template matching by normalized cross correlation (NCC) is widely used for finding image correspondences. NCCNet improves the robustness of this algorithm by transforming image features with siamese convolutional nets trained to maximize the contrast between NCC values of true and false matches. The main technical contribution is a weakly supervised learning algorithm for the training. Unlike fully supervised approaches to metric learning, the method can improve upon vanilla NCC without receiving locations of true matches during training. The improvement is quantified through patches of brain images from serial section electron microscopy. Relative to a parameter-tuned bandpass filter, siamese convolutional nets significantly reduce false matches. The improved accuracy of the method could be essential for connectomics, because emerging petascale datasets may require billions of template matches during assembly. Our method is also expected to generalize to other computer vision applications that use template matching to find image correspondences.

In conclusion, we propose a combination of methods mentioned above to solve large-scale biomedical image processing tasks using distributed deep learning. We obtain the following results.

1. We demonstrate the scalability of the framework by running basic evaluation tasks such as pre-processing, distributed training, hyperparameter search and large-scale inference tasks utilizing 10000 CPU cores and 300 GPU instances with overall processing power of 30 Petaflops.

2. Scalable array and other data storage. It lets distributed neural network optimization from centralized storage. We demonstrate that the distributed file system is equivalent to local file system in terms of training Neural Networks, however it can manage up to peta-scale datasets.

[17]Zlateski, Aleksandar, and H. Sebastian Seung. "Compile-time optimized and statically scheduled ND convnet primitives for multi-core and many-core (Xeon Phi) CPUs." In Proceedings of the International Conference on Supercomputing, p. 8. ACM, 2017.

3. CPU-only deep learning inference library specialized for Intel processors. The library is 3-4x faster compared to widely used Tensorflow and Pytorch. Using the library on CPU instances, it is possible to 1.5x reduce cloud computing costs compared to using GPUs.

4. Weakly-supervised deep metric learning method for template matching. The method improves template matching error rate by 2-8 times compared to existing methods used state-of-the-art pipelines.

5. New deep learning layer that computes channel-wise NCC. Deep Normalized Cross Correlation layer computes cosine similarity per patch instead of euclidean distance and has FFT implementation for large kernels.

Резюме

Для обучения и применения моделей глубокого обучения в реальных приложениях требуется обработка больших объемов данных. Это сложная задача, когда объем данных увеличивается до масштаба в сотни терабайт или даже петабайт. Приложения включают в себя автомобили с автоматическим управлением, анализ аэрофотоснимков, обнаружение ошибок для промышленного производства и обработку биомедицинских изображений. Биомедицинская обработка изображений связана с большими объемами данных, где метки часто недоступны. Изображения могут быть искажены, повреждены и содержать дефекты.

Мы представляем гибридную распределенную облачную инфраструктуру с унифицированным представлением о множестве облаков и локальной инфраструктурой для обработки задач с использованием вычислительных процессоров как CPU, так и GPU в масштабе. Система реализует распределенную файловую систему и отказоустойчивый планировщик обработки задач, независимо от используемого языка или среды глубокого обучения. Это позволяет нам использовать нестабильные дешевые ресурсы в облаке, чтобы значительно сократить расходы на обработку биомедицинских изображений.

В дополнение к оптимизации по вычислительным узлам, мы представляем фреймворк вывода глубоких моделей только для ЦП, который позволяет эффективно обрабатывать данные на одной машине. Движок вывода расширяет ZnnPhi[18], представив JIT-компилятор и дополнительные глубокие слои. Во время компиляции фреймворк объединяет слои, оптимизирует структуру памяти и использует SIMD-инструкции для быстрой обработки нейронных сетей.

Мы представляем слабо контролируемый подход к обучению метрикам, чтобы найти шаблоны в срезах биологического образца, что значительно повышает точность выравнивания. Сопоставление шаблонов с помощью нормализованной взаимной корреляции (NCC) широко используется для поиска соответствий изображений. NCCNet повышает надежность этого алгоритма путем преобразования характеристик изображения с помощью сиамских сверточных сетей, обученных максимизировать контраст между значениями NCC истинных и ложных совпадений. Основным техническим вкладом является слабо контролируемый алгоритм для обучения. В отличие от полностью контролируемых подходов к изучению метрик, метод может улучшить ванильный NCC без получения мест истинных совпадений во время обучения. Улучшение количественно определяется с помощью патчей изображений головного мозга из серийной электронной микроскопии. Относительно настроенного на параметры полосового фильтра сиамские сверточные сети значительно уменьшают ложные совпадения. Повышенная точность метода может быть существенной для коннектомики, потому что новые наборы данных могут потребовать миллиарды совпадений шаблонов во время сборки. Также ожидается, что наш метод будет распространен на другие приложения компьютерного зрения, которые используют сопоставление с шаблоном для поиска соответствий изображений.

[18]Zlateski, Aleksandar, and H. Sebastian Seung. "Compile-time optimized and statically scheduled ND convnet primitives for multi-core and many-core (Xeon Phi) CPUs." In Proceedings of the International Conference on Supercomputing, p. 8. ACM, 2017.

В заключение мы предлагаем комбинацию методов, упомянутых выше, для решения крупномасштабных задач биомедицинской обработки изображений с использованием распределенного глубокого обучения. Получаем следующие результаты.

1. Мы демонстрируем масштабируемость фреймворка, выполняя основные задачи определение качества, такие как предварительная обработка, распределенное обучение, поиск гипер-параметров и крупномасштабные задачи вывода с использованием 10 000 ядер ЦП и 300 экземпляров графического процессора с общей вычислительной мощностью 30 петафлопс.

2. Мы представляем масштабируемый массив и другое хранилище данных. Это позволяет распределенной оптимизации нейронной сети из централизованного хранилища. Мы демонстрируем, что распределенная файловая система эквивалентна локальной файловой системе с точки зрения обучения нейронных сетей, однако она может управлять вплоть до петабайт наборов данных.

3. Библиотека глубокого обучения, предназначенная только для Intel процессора. Библиотека в 3-4 раза быстрее по сравнению с широко используемыми Tensorflow и Pytorch. Используя библиотеку на экземплярах ЦП, можно в 1,5 раза снизить затраты на облачные вычисления по сравнению с использованием графических процессоров.

4. Метод глубокого обучения со слабым контролем для сопоставления для биомедицинских изображений. Метод уменьшает частоту ошибок сопоставления шаблонов в 2-8 раз по сравнению с существующими методами, используемыми в современных системах.

5. Новый слой глубокого обучения, который вычисляет NCC. Слой глубокой нормализации взаимной корреляции вычисляет косинусное сходство для каждого патча, а не евклидово расстояние, и имеет реализацию FFT для больших входных изображений.