

SUN'IY INTELLEKT UCHUN DASTURIY MUHITLAR VA KUTUBXONALARNI O'RNATISH

Tojimamatov Israil Nurmamatovich

Fargʻona davlat universiteti Amaliy matematika
va informatika kafedrası katta oʻqituvchisi

E-mail: israiltojimatov@gmail.com

Zarnigor Rahmiddinova Muzaffarjon qizi

Fargʻona davlat universiteti Amaliy matematika
yoʻnalishi 3-bosqich 23.08-guruh talabasi

E-mail: rahmiddinovazarnigor7@gmail.com

Annotatsiya: Ushbu ilmiy maqolada sun'iy intellekt, mashinaviy o'qitish va chuqur o'rganish sohalarida samarali tadqiqotlar olib borish va amaliy loyihalarni amalga oshirish uchun zarur bo'lgan dasturiy muhitlar va fundamental kutubxonalarni o'rnatishning tizimli va ilmiy asoslari tahlil qilinadi. Tadqiqotda Python dasturlash tilining sun'iy intellekt ekotizimidagi markaziy o'rni yoritiladi va asosiy freymvorklar, jumladan, TensorFlow, PyTorch va Scikit-learn kabi muhim vositalarni turli operatsion tizimlar (masalan, Windows, Linux) ostida konfiguratsiya qilish usullari nazariy jihatdan asoslanadi. Maqolada optimal ish faoliyatiga erishish uchun virtual muhitlar yaratishning ahamiyati, grafik ishlov berish qurilmalari (GPU) uchun drayverlar va CUDA/cuDNN kabi parallel hisoblash texnologiyalarini integratsiya qilishning nozik jihatlari batafsil ko'rib chiqiladi. Ushbu maqola yosh mutaxassislar va talabalar uchun sun'iy intellekt sohasiga kirishda dasturiy ta'minot infratuzilmasini to'g'ri va ilmiy yondashuv bilan qurish bo'yicha mustahkam qo'llanma bo'lib xizmat qiladi.

Kalit so'zlar: Sun'iy intellekt, dasturiy muhit, Python, TensorFlow, PyTorch, Scikitlearn, virtual muhit, GPU, konfiguratsiya.

Аннотация: В данной научной статье проводится системный анализ и научное обоснование установки программного обеспечения и фундаментальных библиотек, необходимых для проведения эффективных исследований и реализации практических проектов в области искусственного интеллекта, машинного обучения и глубокого обучения. В исследовании освещается центральная роль языка программирования Python в экосистеме искусственного интеллекта и теоретически обосновываются методы конфигурации ключевых фреймворков, включая TensorFlow, PyTorch и Scikit-learn, в различных операционных системах (например, Windows, Linux). В статье подробно рассматриваются тонкости интеграции технологий параллельных вычислений, таких как драйверы GPU и CUDA/cuDNN, а также подчеркивается важность создания виртуальных сред для достижения оптимальной производительности. Эта статья послужит надежным руководством для молодых специалистов и студентов по построению инфраструктуры программного обеспечения для искусственного интеллекта с правильным и научным подходом.

Ключевые слова: Искусственный интеллект, программная среда, Python, TensorFlow, PyTorch, Scikit-learn, виртуальная среда, GPU, конфигурация.

Abstract: This scientific article provides a systematic analysis and theoretical substantiation of the installation of necessary software environments and fundamental libraries required for conducting effective research and implementing practical projects in the fields of artificial intelligence, machine learning, and deep learning. The research illuminates the central role of the Python programming language within the AI ecosystem and theoretically justifies methods for configuring key frameworks, including TensorFlow, PyTorch, and Scikit-learn, across various operating systems (e.g., Windows, Linux). The paper meticulously examines the critical aspects of integrating parallel computing technologies, such as GPU drivers and CUDA/cuDNN, and emphasizes the importance of establishing virtual environments to ensure optimal performance. This article serves as a

robust guide for young professionals and students on building the software infrastructure for artificial intelligence with a correct and scientifically informed approach.

Keywords: Artificial intelligence, software environment, Python, TensorFlow, PyTorch, Scikit-learn, virtual environment, GPU, configuration.

Kirish

Sun'iy intellekt texnologiyalarining misli ko'rilmagan darajada rivojlanishi, zamonaviy ilmiy va amaliy tadqiqotlarning markaziy yo'nalishiga aylandi. Ushbu sohada muvaffaqiyatga erishish uchun nazariy bilimlar bilan bir qatorda, yuqori samaradorlikni ta'minlovchi to'g'ri konfiguratsiya qilingan dasturiy muhitga ega bo'lish fundamental ahamiyat kasb etadi. Murakkab neyron tarmoqlarni o'qitish va katta hajmdagi ma'lumotlarga ishlov berish yuqori hisoblash quvvatini talab qiladi, bu esa dasturiy infratuzilmani ilmiy asoslangan holda qurishni taqozo etadi.

Ushbu maqolaning maqsadi – sun'iy intellekt bo'yicha mutaxassislar va tadqiqotchilar uchun zarur bo'lgan barcha dasturiy komponentlarni, ya'ni dasturlash tilidan tortib to optimallashtirilgan chuqur o'rganish freymvorklarigacha bo'lgan to'plamni o'rnatish va konfiguratsiya qilishning nazariy va amaliy jihatlarini tizimli ravishda tahlil qilishdan iborat. Biz ushbu jarayonni faqatgina texnik ko'rsatma sifatida emas, balki ilmiy samaradorlikni oshirish strategiyasi sifatida ko'rib chiqamiz.

Dastlab, Python dasturlash tilining AI ekotizimidagi ajralmas o'rni, so'ngra ma'lumotlarga ishlov berish, mashinaviy o'qitish va chuqur o'rganish uchun mo'ljallangan asosiy kutubxonalarning nazariy asoslari ko'rib chiqiladi. Shuningdek, turli loyihalarning bog'liqliklarida (dependencies) kelib chiqadigan muammolarning oldini olish uchun virtual muhitlar yaratishning ilmiy zarurati asoslanadi. Nihoyat, hisoblash tezligini keskin oshiruvchi GPU texnologiyalarini operatsion tizimga muvaffaqiyatli integratsiya qilish usullari chuqur tahlil qilinadi. Ushbu maqola AI loyihalarining poydevorini mustahkamlashga qaratilgan bo'lib, xalqaro ilmiy hamjamiyatning yuqori standartlariga javob beradi.

Asosiy qism

Sun'iy intellekt sohasidagi deyarli barcha ilmiy tadqiqotlar va amaliy ishlanmalar Python dasturlash tili atrofida jamlangan bo'lib, bu tilning ushbu ekotizimdagi o'rni beqiyosdir. Pythonning ushbu mavqeyi uning sintaksisining soddaligi, o'rganishning qulayligi va eng asosiysi, keng va boy kutubxonalar to'plami mavjudligi bilan izohlanadi.

Pythonning ilmiy hisoblashlar uchun mosligi uning Numpy kabi asosiy kutubxonalari bilan bog'liqdir. Numpy ko'p o'lchamli massivlar va matritsalar ustida yuqori tezlikda arifmetik amallarni bajarish imkonini beradi. Neyron tarmoqlar nazariy jihatdan aynan matritsalar algebrasi va vektor amallariga asoslangan bo'lib, Numpy ushbu amallarni S yoki Fortran kabi past darajali tillarda optimallashtirilgan holda taqdim etadi. Ushbu fundamental vositaning o'rnatilishi har qanday AI muhitini yaratishning birinchi va eng muhim bosqichi hisoblanadi.

Bundan tashqari, Python ma'lumotlarga ishlov berish uchun Pandas (ma'lumotlar ramkalari), ilmiy va statistik hisoblashlar uchun Scipy kabi kuchli kutubxonalarni o'zida birlashtirgan. Ushbu kutubxonalarning standart Python muhitiga o'rnatilishi ma'lumotlarni tahlil qilish, oldindan qayta ishlash va vizuallashtirish kabi ilmiy tadqiqot bosqichlarini samarali bajarishga imkon yaratadi. Shuning uchun, dasturiy muhitni o'rnatish jarayoni to'g'ri versiyadagi Python tarjimonini operatsion tizimga o'rnatishdan va uning paket menejeri (pip) orqali yuqorida sanalgan asosiy kutubxonalarni yuklab olishdan boshlanishi ilmiy jihatdan asoslangan yondashuvdir.

Sun'iy intellekt tadqiqotlari uchun freymvorklarni to'g'ri o'rnatish, optimal ish faoliyati va loyihalararo barqarorlikni ta'minlashda hal qiluvchi ahamiyatga ega. Ushbu jarayonni amalga oshirishda ikki asosiy guruhdagi kutubxonalarga e'tibor qaratiladi: mashinaviy o'qitish va chuqur o'rganish freymvorklari.

Scikit-learn kutubxonasi klassik mashinaviy o'qitish algoritmlari (masalan, regressiya, tasniflash va klasterlash) uchun eng keng tarqalgan va barqaror vosita hisoblanadi. Uning o'rnatilishi nisbatan sodd bo'lib, asosan pip paket menejeri orqali

amalgaga oshiriladi va u Numpy hamda Scipy kabi fundamental ilmiy kutubxonalarga bevosita bog'liqdir. Scikit-learnning to'g'ri o'rnatilishi, dasturiy muhitda ma'lumotlar tahlili va dastlabki modellashtirish ishlarini bajarish uchun asosiy shartdir.

Chuqur o'rganish (Deep Learning) uchun mo'ljallangan TensorFlow va PyTorch kabi freymvorklar murakkab hisoblash grafigi (computational graph) tuzilishiga ega bo'lgan neyron tarmoqlarni qurish uchun ishlatiladi. Ushbu freymvorklarning o'rnatilishi bir necha omillarni hisobga olishni talab qiladi:

Markaziy Prosessor (CPU) Versiyasi: Dastlab, ushbu freymvorklarning faqat CPU yordamida ishlaydigan standart versiyalari o'rnatiladi. Ushbu versiyalar nisbatan oddiy loyihalar va kodni sinovdan o'tkazish uchun yetarli hisoblanadi.

GPU Integratsiyasi Versiyasi: Loyihalar katta hajmdagi ma'lumotlar va murakkab neyron tarmoqlarni talab qilganida, freymvorklarning GPU yordamida ishlash uchun optimallashtirilgan versiyalarini o'rnatish zarur bo'ladi. Bu esa CUDA va cuDNN kabi NVIDIA texnologiyalarining versiyalari bilan freymvork versiyalari orasidagi qattiq muvofiqlikni ta'minlashni talab qiladi.

Bu bosqichda har bir freymvorkning aniq talablariga muvofiq Python versiyasini tanlash ilmiy-texnik jihatdan muhim hisoblanadi, chunki noto'g'ri versiyalar bog'liqliklar o'rtasida ziddiyatlarga olib kelishi va ish faoliyatini sezilarli darajada pasaytirishi mumkin.

Ilmiy tadqiqotlarda va dasturiy loyihalashda virtual muhitlar (Virtual Environments) yaratish va ulardan foydalanish shart hisoblanadi. Virtual muhitlar bir necha muhim ilmiytexnik muammolarni hal etadi:

Birinchidan, Bog'liqliklar Ziddiyatini Oldini Olish. Har bir sun'iy intellekt loyihasi turli xil kutubxona versiyalarini talab qilishi mumkin. Masalan, bir loyiha TensorFlowning Eskiroq Versiyasini talab qilsa, boshqasi PyTorchning Yangi Versiyasini talab qilishi mumkin. Agar barcha kutubxonalar bir global muhitga o'rnatilsa, bu versiya ziddiyatlari dasturlarning to'g'ri ishlashiga xalaqit beradi. Virtual muhit har bir loyiha uchun alohida, izolyatsiya qilingan Python muhitini yaratib, ushbu ziddiyatni butunlay bartaraf etadi.

Ikkinchidan, Takrorlanuvchanlikni Ta'minlash (Reproducibility). Ilmiy tadqiqotlar natijalari har qanday muhitda aynan takrorlanishi kerak. Virtual muhit loyihada ishlatilgan barcha kutubxonalar va ularning aniq versiyalarini ro'yxatga oluvchi fayl (masalan, requirements.txt) yaratish imkonini beradi. Bu fayl boshqa tadqiqotchilarga loyihani aynan bir xil muhitda qayta tiklashga imkon beradi, bu esa ilmiy ishlarning validligi (haqqoniyligi)ni oshiradi.

Ushbu muhitlarni yaratish uchun asosan venv yoki Conda kabi vositalardan foydalaniladi. Conda kabi vositalar nafaqat Python kutubxonalarini, balki Python tarjimonining o'zini ham turli versiyalarda boshqarishga imkon berib, ayniqsa chuqur o'rganish kabi murakkab sohalarda katta qulaylik yaratadi. Virtual muhitni to'g'ri boshqarish sun'iy intellekt bo'yicha professional mutaxassisning ajralmas amaliyoti hisoblanadi.

GPU Va Parallel Hisoblash Texnologiyalarini Integratsiyalash

Chuqur o'rganish modellarini o'qitishda hisoblash quvvatiga bo'lgan talab juda yuqori bo'lib, Grafik Ishlov Berish Qurilmalari (GPU) dan foydalanish standart ****Markaziy Prosessorlar (CPU)****ga nisbatan o'qitish vaqtini bir necha barobar qisqartirish imkonini beradi. GPUning asosiy afzalligi uning parallel hisoblash qobiliyatiga ega bo'lishi bilan izohlanadi, bu esa neyron tarmoqlarning matritsa amallarini tezkor bajarish uchun idealdir.

GPU ni AI muhitiga muvaffaqiyatli integratsiya qilish ilmiy jihatdan aniq va izchil qadamlarni talab qiladi:

NVIDIA Drayverlarini O'rnatish: Birinchi navbatda, ishlatilayotgan NVIDIA GPU uchun operatsion tizimga mos keladigan eng so'nggi va barqaror drayverlar o'rnatilishi shart. Noto'g'ri yoki eskirgan drayverlar GPU dan foydalanishda jiddiy ishlash muammolariga olib keladi.

CUDA Toolkit O'rnatilishi: CUDA (Compute Unified Device Architecture) bu NVIDIA tomonidan ishlab chiqilgan parallel hisoblash platformasi va dasturlash modeli bo'lib, u CPU va GPU o'rtasida aloqani ta'minlaydi. TensorFlow va PyTorch kabi

freymvorklarning GPU versiyalari ishlashi uchun mos versiyadagi CUDA Toolkit o'rnatilgan bo'lishi talab etiladi. Bu bosqichda eng katta ilmiy-texnik muammo freymvorkning aniq talab qilinadigan CUDA versiyasi bilan kompyuterga o'rnatilgan CUDA versiyasi o'rtasida to'liq moslikni ta'minlashdir.

cuDNN Kutubxonasini Integratsiyalash: cuDNN (CUDA Deep Neural Network library) bu neyron tarmoqlar uchun optimallashtirilgan tezlatilgan primitiv operatsiyalarni o'z ichiga olgan kutubxonadir. U konvolyutsiya, pooling va normalizatsiya kabi operatsiyalarni yuqori samaradorlikda bajarish uchun CUDAgga qo'shimcha tarzda o'rnatiladi. cuDNNni freymvork va CUDA versiyasi bilan to'g'ri birlashtirish chuqur o'rganish modellarining o'qitish samaradorligini keskin oshirishga xizmat qiladi.

Ushbu integratsiya bosqichlari har bir freymvorkning rasmiy hujjatlariga qat'iy rioya qilgan holda amalga oshirilishi shart, aks holda dasturiy muhit GPU quvvatidan foydalana olmaydi va o'qitish jarayoni asossiz ravishda uzayadi.

Dasturiy muhit muvaffaqiyatli o'rnatilgandan so'ng, keyingi muhim qadam amaliy ish oqimini (workflow) ilmiy jihatdan asoslangan holda tashkil etishdan iborat. Bu oqim sun'iy intellekt loyihalarining har bir bosqichida o'rnatilgan kutubxonalar va muhitlardan maksimal darajada foydalanishni ta'minlashi lozim.

Ideal ish oqimi ma'lumotlarni yuklashdan to modelni baholashgacha bo'lgan to'rt bosqichli izchillikka asoslanadi. Dastlabki bosqich ma'lumotlarni yig'ish va yuklash bo'lib, bu yerda Pandas kabi vositalar yordamida ma'lumotlar virtual muhitga kiritiladi. Ikkinchi bosqich ma'lumotlarni oldindan qayta ishlash (preprocessing) bo'lib, bunda Scikit-learn va Numpy funksiyalari ma'lumotlarni tozalash, transformatsiya qilish va normallashtirish uchun qo'llaniladi. Uchinchidan, modelni qurish va o'qitish bosqichi keladi, bunda TensorFlow yoki PyTorch ning GPU yordamida tezlashtirilgan versiyalari aktivlashtirilib, neyron tarmoq massiv ma'lumotlar ustida o'qitiladi. Nihoyat, to'rtinchi bosqich modelni baholash va tekshirish bo'lib, ushbu jarayonda olingan natijalarning ilmiy asoslanganligini ta'minlash uchun xatolik ko'rsatkichlari tahlil qilinadi. Ushbu tizimli ish oqimi tadqiqot natijalarining haqqoniyligini (validity) oshiradi.

Eng murakkab ilmiy-texnik qiyinchiliklardan biri kutubxonalar versiyalari o'rtasidagi ziddiyatdir. Bunday ziddiyatlar, asosan, asosiy freymvork (masalan, TensorFlow) bilan bog'liq bo'lgan past darajadagi kutubxonalar (masalan, Numpy yoki Keras) o'rtasida moslikning buzilishi natijasida yuzaga keladi. Bu muammoni hal qilish strategiyasi loyiha uchun zarur bo'lgan har bir freymvorkning rasmiy talablarini chuqur tahlil qilish va virtual muhitni ularning eng konservativ umumiy versiyalari asosida konfiguratsiya qilishni o'z ichiga oladi. Ayniqsa, chuqur o'rganish freymvorklari va ularning CUDA talablari doimiy ravishda o'zgarib turganligi sababli, virtual muhitni yaratishdan oldin rasmiy versiya matritsasini tekshirish ilmiy yondashuvning asosiy qismidir.

AI tadqiqotlari doimo turli hisoblash uskunalarida o'tkaziladi, shuning uchun dasturiy muhitning apparat ta'minotiga moslashish strategiyasi muhimdir. GPU mavjud bo'lmagan hollarda, freymvorkni CPU yordamida ishlashga to'g'ri sozlash va multiprocessing kabi usullardan foydalanib hisoblashni parallellashtirish ilmiy samaradorlikni saqlab qolishga yordam beradi. GPU mavjud bo'lganda esa, ma'lumotlarni CPU xotirasidan GPU xotirasiga o'tkazishni optimallashtirish orqali o'qitish jarayonidagi bottleneck (ish unumdorligining pasayishi)ni kamaytirish ilmiy-texnik jihatdan asoslanadi. PyTorch kabi freymvorklar bu jarayonni avtomatlashtirish uchun qulay interfeyslar taqdim etadi, biroq ulardan foydalanish uchun dasturiy muhitning konfiguratsiyasi yuqori darajada aniq bo'lishi shart.

O'rnatish jarayonida yuzaga keladigan xatoliklar odatda drayverlar, CUDA, cuDNN va freymvork versiyalari o'rtasidagi nomuvofiqlikdan kelib chiqadi. Eng keng tarqalgan xatoliklardan biri bu freymvorkning "GPU ni topa olmasligi" xabaridir. Bunday hollarda, muammoni hal qilishning ilmiy metodi tizimli tekshiruvga asoslanadi: avval drayverlar to'g'ri o'rnatilganligini, so'ngra freymvork uchun talab qilinadigan CUDA va cuDNN versiyalari mavjudligini va oxirida virtual muhitda freymvorkning GPU versiyasi to'g'ri chaqirilayotganligini tekshirish lozim. Barcha komponentlarning aniq versiyalarini tekshirish va ularning rasmiy hujjatlar talablariga mosligini qiyosiy tahlil qilish, muammoni tezkor aniqlashning eng ishonchli usuli hisoblanadi.

Xulosa

Ushbu ilmiy maqola sun'iy intellekt tadqiqotlari uchun dasturiy muhitlar va kutubxonalarni o'rnatish jarayonining nazariy asoslarini va amaliy metodologiyasini tizimli ravishda tahlil qildi. Tadqiqotlar shuni ko'rsatdiki, sun'iy intellektning muvaffaqiyati faqatgina algoritmlarning murakkabligiga emas, balki ularni bajarish uchun mo'ljallangan infratuzilmaning ilmiy asoslangan holda konfiguratsiya qilinganligiga ham bog'liqdir.

Birinchidan, Python tili, uning Numpy va Pandas kabi fundamental kutubxonalari bilan birgalikda, sun'iy intellekt uchun ajralmas poydevor hisoblanadi. Ikkinchidan, TensorFlow va PyTorch kabi chuqur o'rganish freymvorklarini o'rnatishda ularning GPU texnologiyalari bilan muvofiqligini ta'minlash muhimdir. Uchinchidan, virtual muhitlardan foydalanish loyihalararo bog'liqliklar ziddiyatini oldini olish va ilmiy natijalarning takrorlanuvchanligini kafolatlash uchun shart bo'lgan ilmiy amaliyotdir. Va nihoyat, CUDA va cuDNN ni drayverlar bilan to'g'ri integratsiya qilish orqali erishiladigan parallel hisoblash imkoniyati murakkab chuqur o'rganish modellarini o'qitish tezligini oshirishda hal qiluvchi omil bo'lib xizmat qiladi.

Foydalanilgan Adabiyotlar

1. Goodfellow, I., Bengio, Y., & Courville, A. (Ikki Ming O'n Yetti). Deep Learning. MIT Press nashriyoti. (Chuqur o'rganishning nazariy asoslari, matritsa amallari va hisoblash grafigining ilmiy-texnik tahlili uchun.)
2. McKinney, W. (Ikki Ming O'n Ikki). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media nashriyoti. (Pandas va NumPy kabi fundamental ma'lumotlarga ishlov berish kutubxonalarining nazariy asoslari va Python ekotizimidagi o'rni uchun.)
3. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Isard, M. (Ikki Ming O'n Sez). TensorFlow: A System for Large-Scale Machine Learning. O'n ikkinchi USENIX Simpoziumi. (TensorFlow freymvorkining arxitekturasini, CPU va GPU da ishlash prinsiplari bo'yicha ilmiy manba.)

4. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (Ikkki Ming O'n To'qqiz). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems nashriyoti. (PyTorch kutubxonasining dinamik hisoblash grafigi va uning yuqori samaradorlikka erishish mexanizmlari bo'yicha ilmiy manba.)
5. Tojimamatov, I. N., & Saidjamolova, B. M. (2023). BIZNESDA «BIG DATA» TEXNOLOGIYALARI VA ULARNING AHAMIYATI. Лучшие интеллектуальные исследования, 11(4), 56–63.
6. Tojimamatov, I. N., & Azizjon o'g'li, N. A. Z. (2024). The SQL server language and its structure. American Journal of Open University Education, 1(1), 11–15.
7. Tojimamatov, I. N., Topvoldiyeva, H., Karimova, N., & Inomova, G. (2023). GRAFIK MA'LUMOTLAR BAZASI. Евразийский журнал технологий и инноваций, 1(4), 75–84.
8. Tojimamatov, I. N., & Gulhayo, M. (2023). MA'LUMOTLARNI QAYTA ISHLASHDA ERP TIZIMLARI. MA'LUMOTLARNI QAYTA ISHLASHDA SAP TIZIMLARI. Journal of Integrated Education and Research, 2(4), 87–89.