

KERAS FREYMVORKI: O'RNATISH VA ASOSIY FUNKSIYALAR.

Tojimamatov Israil Nurmamatovich

*Farg'ona davlat universiteti Amaliy matematika
va informatika kafedrası katta o'qituvchisi*

E-mail: israiltojimatov@gmail.com

Shahzoda O'rinboyeva Mahammadjon qizi

*Farg'ona davlat universiteti Amaliy matematika
yo'nalishi 3-bosqich 23.08-guruh talabasi*

E-mail: shahzodaorinboyeva@gmail.com

Annotatsiya: Ushbu ilmiy maqola Keras freymvorki ning chuqur o'rganish modellarini loyihalash, o'qitish va baholashdagi markaziy ahamiyatini tahlil qilishga bag'ishlangan. Maqola Kerasning yuqori darajadagi abstraksiya qatlami sifatida qo'llanilishi, uning TensorFlow (yoki boshqa bekkendlar) bilan integratsiyasi va sodda, modulli arxitekturası tufayli ilmiy tadqiqotlar va sanoat loyihalarida keng tarqalishining asosiy sabablarini ko'rib chiqadi. Kerasning o'rnatish jarayonlari, asosiy funksional imkoniyatlari, jumladan, Sequential va Functional API orqali neyron tarmoqlarini qurish, optimizatorlar, yo'qotish funksiyalari va metrikalarni sozlash mexanizmlari ilmiy tilda chuqur yoritilgan. Shuningdek, modelni saqlash, yuklash va o'qitish jarayonini nazorat qiluvchi funksiyalarga alohida e'tibor qaratilgan. Tadqiqot chuqur o'rganish sohasi mutaxassislari, o'qituvchilari va talabalari uchun Kerasning nazariy va amaliy jihatlarini anglashga yordam beradi.

Kalit So'zlar: Keras, Chuqur O'rganish, Neyron Tarmoq, Sequential API, Functional API, Abstraksiya Qatlami, O'rnatish, Modellashtirish.

Аннотация: Данная научная статья посвящена анализу центральной роли фреймворка Keras в проектировании, обучении и оценке моделей глубокого обучения. В статье рассматриваются ключевые причины широкого распространения Keras в научных исследованиях и промышленных проектах, обусловленные его использованием в качестве высокоуровневого уровня абстракции, его интеграцией с TensorFlow (или другими бэкендами) и его простой, модульной архитектурой. Подробно и научно освещены процессы установки Keras, его основные функциональные возможности, включая построение нейронных сетей с использованием Sequential и Functional API, а также механизмы настройки оптимизаторов, функций потерь и метрик. Особое внимание уделяется функциям сохранения, загрузки и контроля процесса обучения модели. Исследование поможет специалистам, преподавателям и студентам в области глубокого обучения понять как теоретические, так и практические аспекты Keras.

Ключевые Слова: Keras, Глубокое Обучение, Нейронная Сеть, Sequential API, Functional API, Уровень Абстракции, Установка, Моделирование.

Abstract: This scientific article is dedicated to analyzing the central role of the Keras framework in the design, training, and evaluation of deep learning models. The paper examines the key reasons for Keras's widespread adoption in scientific research and industrial projects, driven by its use as a high-level abstraction layer, its seamless integration with TensorFlow (and other backends), and its simple, modular architecture. The article scientifically and thoroughly covers Keras installation procedures and its core functional capabilities, including the construction of neural networks using the Sequential and Functional API, as well as the mechanisms for configuring optimizers, loss functions, and metrics. Special attention is given to functions for saving, loading, and managing the model training process. This research serves to assist professionals, educators, and students in the field of deep learning in comprehending both the theoretical and practical aspects of Keras.

Keywords: Keras, Deep Learning, Neural Network, Sequential API, Functional API, Abstraction Layer, Installation, Modeling.

KIRISH

So'nggi o'n yillikda sun'iy intellekt va mashinada o'rganish sohaslarida chuqur o'rganish (Deep Learning) modellari misli ko'rilmagan yutuqlarga erishdi, bu esa tabiiy tilni qayta ishlash, tasvirlarni tanib olish va robototexnika kabi ko'plab sohalarda inqilobiy o'zgarishlarga sabab bo'ldi. Bu yutuqlarning markazida murakkab neyron tarmoqlarni samarali loyihalash va o'qitish imkonini beruvchi dasturiy ta'minot freymvorklari turadi. Keras – bu Python tilida yozilgan, yuqori darajadagi, modulli va kengayishga ochiq bo'lgan neyron tarmoq freymvorkidir. U dastlab abstraksiya qatlami sifatida ishlab chiqilgan bo'lib, past darajadagi bekkend hisoblash kutubxonalari (dastlab Theano va Kognitiv Toolkit, hozirda esa asosan TensorFlow) ustida ishlaydi.

Kerasning asosiy falsafasi aqliy zo'riqishni minimallashtirish va tezkor eksperimentatsiyani ta'minlashdir. Bu uni nafaqat ilmiy tadqiqotchilar va talabalar orasida, balki tezkor prototiplashni talab qiladigan sanoat muhitlarida ham eng mashhur vositalardan biriga aylantirdi. Keras murakkab chuqur o'rganish arxitekturalarini juda kam kod bilan qurish imkonini beradi, shu bilan birga modullarning to'liq nazoratini saqlab qoladi.

Ushbu ilmiy maqolaning maqsadi Keras freymvorkining fundamental jihatlarini, uning ilmiy tadqiqotlardagi ahamiyatini, o'rnatish jarayonining texnologik asoslarini va chuqur o'rganish modellarini yaratish, kompilyatsiya qilish va o'qitish uchun mavjud bo'lgan asosiy funksional imkoniyatlarini batafsil tahlil qilishdan iborat. Ushbu tahlil Kerasning chuqur o'rganish sohasidagi rolini, uning soddaligi va samaradorligi ortidagi ilmiy asoslarni yoritishga qaratilgan.

ASOSIY QISM

Kerasning asosiy ilmiy yutug'i uning yuqori darajali abstraksiya qatlami sifatida ishlashidadir. Kerasning kodi bevosita past darajadagi matritsa operatsiyalari yoki gradient hisob-kitoblari bilan shug'ullanmaydi. Aksincha, u bekkend freymvorklari (asosan TensorFlow) ustida joylashgan bo'lib, uning funksiyalarini past darajadagi amalga oshirishni bekkendga yuklaydi. Bu ikki darajali arxitektura foydalanuvchiga murakkab chuqur o'rganish arxitekturalarini intuitiv Python sintaksisi yordamida loyihalash imkonini beradi, shu bilan birga zamonaviy hisoblash optimallashtirishlaridan (masalan, GPU va TPU dagi parallel hisoblash) foydalanish imkoniyatini saqlab qoladi.

Kerasning bunday modulli tuzilishi uning ilmiy tadqiqotlarda qo'llanilishini soddalashtiradi, chunki tadqiqotchilar model arxitekturasini va eksperiment dizayniga e'tibor qaratishi mumkin, past darajadagi Tensor manipulyatsiyasi va avtomatik differensiallashning murakkabliklaridan xalos bo'ladi. Bekkend sifatida TensorFlow ni ishlatish Kerasga avtomatik differensiallash (Automatic Differentiation) imkoniyatini beradi, bu esa neyron tarmoqlarni o'qitishda asosiy bo'lgan teskari tarqalish (Backpropagation) algoritmini samarali amalga oshirish uchun zarurdir. Kerasning bekkenddan mustaqilligi uning freymvorklar ekotizimidagi moslashuvchanligini ta'minlaydi.

Keras freymvorkini ilmiy va eksperimental maqsadlar uchun o'rnatish odatda Python muhiti menejeri (masalan, Conda yoki venv) yordamida amalga oshiriladi. Eng muhim qadam, Keras bilan integratsiyalangan bekkend kutubxonasini, odatda TensorFlow ni o'rnatishdir. Ilmiy ishlar va katta hajmli ma'lumotlar bilan ishlash uchun GPU (Grafik Ishlov Berish Qurilmasi) drayverlari va uning hisoblash API'si, masalan, CUDA (Compute Unified Device Architecture) va cuDNN (CUDA Deep Neural Network Library) ni to'g'ri o'rnatish talab etiladi.

GPU lardan foydalanish chuqur o'rganish modellarini o'qitish vaqtini astronomik darajada qisqartiradi, chunki bu neyron tarmoqlaridagi matritsa ko'paytirish operatsiyalarini yuqori darajada parallel bajarish imkonini beradi. Shuning uchun, ilmiy maqolalarda Kerasdan foydalanishni hujjatlashtirishda, nafaqat freymvorkning versiyasi, balki ishlatilgan hisoblash muhitining (CPU/GPU) konfiguratsiyasi ham aniq ko'rsatilishi zarur, chunki bu natijalarning qayta takrorlanuvchanligi (reproducibility) uchun fundamental ahamiyatga ega. O'rnatish jarayonining texnik talablari yuqori unumdorlikdagi hisoblashlar va parallelizatsiya ilmiy me'yorlariga bevosita bog'liqdir.

Keras neyron tarmoq modellarini qurish uchun ikkita asosiy paradigmani taqdim etadi, bu ularning ilmiy murakkablik darajasini boshqarish imkonini beradi. Sequential API eng sodda model qurish usuli bo'lib, unda qatlamlar ketma-ket, chiziqli yig'iladi.

Bu turdagi arxitektura, masalan, oddiy Ko'p Qatlamli Perseptronlar yoki oddiy Ketma-ket Tarmoqlar (Simple Sequential Networks) kabi tekis tuzilmalarni prototiplash uchun idealdir. U minimal kod bilan tezkor loyihalash imkoniyatini beradi.

Biroq, murakkab ilmiy tadqiqotlarda ko'pincha Functional API dan foydalaniladi. Functional API tadqiqotchiga arxitektura ustidan to'liq nazorat beradi va murakkab, chiziqli bo'lmagan tarmoq tuzilmalarini, masalan, bir nechta kirish yoki chiqish qatlamlariga ega bo'lgan modellarni, umumiy qatlamlarni (Shared Layers) yoki rezidual bog'lanishlar (Residual Connections) kabi ilg'or topologiyalarni yaratishga imkon beradi. Rezidual bog'lanishlar gradientning tarmoqning chuqur qatlamlariga to'siqlarsiz o'tishini ta'minlab, chuqur tarmoqlarni o'qitishni barqarorlashtirishda ilmiy jihatdan muhimdir. Functional API neyron tarmoqlarning axborot oqimini grafik model sifatida aniq belgilash imkonini beradi.

Model arxitekturasi aniqlangandan so'ng, u o'qitish jarayoni uchun kompilyatsiya qilinishi kerak. Kompilyatsiya bu modelni o'qitish uchun zarur bo'lgan uchta fundamental komponentni – optimizator, yo'qotish funksiyasi (Loss Function) va metrikalarni belgilashni o'z ichiga oladi.

Optimizator modelning o'qitish jarayonida tarmoq parametrlarini (vaznlarini) minimal darajaga tushirish uchun qanday yangilanishini boshqaradigan algoritm hisoblanadi. Keras ko'plab optimizatorlarni taklif etadi, masalan, Adam, RMSprop va Stoxastik Gradient Tushish (SGD). Adam optimizatori o'zining adaptiv o'rganish tezligi tufayli ilmiy tadqiqotlarda eng ko'p qo'llaniladi. U har bir parametr uchun o'rganish tezligini mustaqil ravishda moslashtiradi.

Yo'qotish funksiyasi modelning bashorati va haqiqiy qiymatlar orasidagi farqni (xatolikni) miqdoriy baholaydi. Masalan, ikkilik tasniflash uchun binary cross-entropy, regressiya vazifalari uchun mean squared error kabi funksiyalar qo'llaniladi. Yo'qotish funksiyasini to'g'ri tanlash modelning konvergensiyasi va umumlashtirish qobiliyati uchun ilmiy jihatdan juda muhimdir.

Metrikalar modelning ish faoliyatini inson tomonidan talqin qilinadigan shaklda baholash uchun ishlatiladi, lekin ular optimizatsiya jarayoniga bevosita ta'sir qilmaydi. Masalan, tasniflashda aniqlik (accuracy) metrikasi qo'llaniladi.

Kerasda modelni o'qitish fit funksiyasi orqali amalga oshiriladi, bu funksiya o'qitish ma'lumotlari, davrlar soni (epochs) va paket hajmi (batch size) kabi asosiy giperparametrlar bilan belgilanadi. Davrlar soni butun o'quv ma'lumotlar to'plamini necha marta tarmoqdan o'tkazishni, paket hajmi esa bir vaqtning o'zida nechta namuna tarmoqqa kiritilishini bildiradi.

O'qitish jarayonini samarali nazorat qilish va ilmiy natijalarni yaxshilash uchun Keras Callback mexanizmini taklif etadi. Callback lar o'qitish davrida avtomatik tarzda bajariladigan funksiyalar to'plamidir. Eng muhim callback larga Early Stopping (keraksiz uzoq o'qitish va overfitting ni oldini oladi) va Model Checkpointing (eng

yaxshi natijalarga erishgan model vaznlarini saqlash) kiradi. Early Stopping o'qitish davrida validatsiya yo'qotishi ma'lum sonli davrlar davomida yaxshilanmasa, o'qitishni to'xtatadi. Bu esa hisoblash resurslarini optimallashtirish va modelning umumlashtirish qobiliyatini oshirishning ilmiy usulidir.

Ilmiy tadqiqotlarda natijalarning qayta takrorlanuvchanligi (reproducibility) fundamental talab hisoblanadi. Keras bu talabni modellar va ularning vaznlarini samarali serializatsiya qilish (saqlash) orqali qo'llab-quvvatlaydi. Modelning butun arxitekturasi, optimizator holati va vaznlari HDF5 kabi universal formatlarda saqlanishi mumkin. Bu tadqiqotchiga o'qitish jarayonini istalgan vaqtda qayta tiklash yoki modelni boshqa tizimga o'tkazish imkonini beradi.

Modelning ish faoliyatini baholash evaluate funksiyasi orqali amalga oshiriladi. Bu funksiya test ma'lumotlari to'plamida yo'qotish funksiyasi va belgilangan metrikalar qiymatlarini hisoblab chiqadi. Natijalarni predict funksiyasi orqali olish mumkin, bu esa kirish ma'lumotlariga modelning bashoratlarini hosil qiladi. Baholash jarayonining shaffofligi Kerasning ilmiy sohadagi ishonchliligini ta'minlovchi muhim omil hisoblanadi. Shuningdek, modelni TensorBoard kabi vizualizatsiya vositalariga integratsiyalash orqali o'qitish jarayonining dinamikasi va tarmoq topologiyasini chuqur tahlil qilish mumkin.

Kerasning moduly dizayni foydalanuvchilarga freymvorkning mavjud imkoniyatlarini kengaytirishga imkon beradi, bu esa eksperimental neyron tarmoq arxitekturalarini loyihalashda ilmiy erkinlikni ta'minlaydi. Tadqiqotchilar o'zlarining maxsus qatlamlari (Custom Layers), yo'qotish funksiyalari (Custom Loss Functions), metrikalari va optimizatorlarini yaratishlari mumkin.

Maxsus qatlamlarni yaratish Kerasning past darajadagi bekkend operatsiyalaridan foydalanishni talab qiladi, bu esa tadqiqotchiga yangi turdagi matematik operatsiyalarni (masalan, yangi turdagi konvolyutsiya yoki pooling operatsiyalari) neyron tarmoqqa kiritish imkonini beradi. Maxsus yo'qotish funksiyalari esa vazifaning o'ziga xos ehtiyojlariga mos keladigan noyob o'qitish maqsadlarini (masalan, aniq sinflarga ko'proq e'tibor qaratish) belgilash imkoniyatini beradi. Kerasning bu kengaytirilish qobiliyati chuqur o'rganish sohasidagi ilmiy innovatsiyalar uchun hal qiluvchi ahamiyatga ega, chunki u mavjud freymvork chegaralaridan tashqariga chiqadigan g'oyalarni amalga oshirishga imkon beradi.

XULOSA

Keras freymvorki chuqur o'rganish sohasida nafaqat foydalanish qulayligi, balki yuqori darajadagi abstraksiya va bekkendlar bilan samarali integratsiyasi tufayli ilmiy va amaliy ahamiyatga ega bo'lgan asosiy vosita sifatida o'z o'rnini mustahkamladi. Uning Sequential va Functional API lari turli murakkablikdagi neyron tarmoqlarni loyihalashda yuqori darajadagi moslashuvchanlikni ta'minlaydi. Optimizatorlar, yo'qotish funksiyalari va Callback mexanizmlari kabi asosiy funksiyalarning yaxshi

tashkil etilganligi tezkor prototiplash va barqaror ilmiy eksperimentlarning o'tkazilishiga imkon yaratadi. Kerasning kengaytirilish imkoniyatlari esa tadqiqotchilarga yangi arxitekturalar va o'qitish usullarini sinovdan o'tkazishda erkinlik beradi. Keras, shubhasiz, chuqur o'rganishning nazariy yutuqlarini amaliy qo'llashga va ilmiy natijalarning qayta takrorlanuvchanligini ta'minlashga yordam beruvchi kuchli texnologik vosita bo'lib qoladi.

FOYDALANILGAN ADABIYOTLAR

1. Chollet, F. (Ikki Ming O'n Yetti). *Python yordamida chuqur o'rganish*. Birinchi nashr. Keras freymvorkining yaratuvchisi tomonidan yozilgan fundamental manba.
2. Abadi, M., et al. (Ikki Ming O'n Besh). TensorFlow: Large-Scale Mashinada O'rganish uchun Ochiq Manbali Dasturiy Ta'minot Kutubxonasi. *Past darajadagi bekkend asoslari bo'yicha ma'lumot*.
3. Goodfellow, I., Bengio, Y., & Courville, A. (Ikki Ming O'n Olti). *Chuqur O'rganish*. Birinchi nashr. *Chuqur o'rganishning nazariy asoslari, optimizatorlar va yo'qotish funksiyalari bo'yicha ma'lumotlar*.
4. Ruder, S. (Ikki Ming O'n Olti). Gradient Tushish Optimizatsiyasiga Qayta Nazar. *ArXiv E-Chop* jurnali. *Zamonaviy optimizatorlarning, masalan, Adamning ilmiy tahlili*.
5. Kingma, D. P., & Ba, J. (Ikki Ming O'n To'rt). Adam: Adaptiv Moment Baholash Usuli. *Xalqaro Konferensiya materiallari. Adam optimizatorini joriy etgan original ish*.
6. Paszke, A., et al. (Ikki Ming O'n To'qqiz). PyTorch: Imperativ uslubdagi Chuqur O'rganish. *Statistik Kompyuter jurnali. Keras bilan raqobatlashuvchi freymvorklar kontekstida abstraksiya qatlamlarini tushunish uchun*.
7. Simonyan, K., & Zisserman, A. (Ikki Ming O'n To'rt). Katta masshtabli tasvir tanib olish uchun juda chuqur konvolyutsion tarmoqlar. *Ilg'or neyron tarmoq arxitekturalarini Functional API orqali amalga oshirishga doir misol*.
8. Tojimamatov, I. N., Topvoldiyeva, H., Karimova, N., & Inomova, G. (2023). GRAFIK MA'LUMOTLAR BAZASI. *Евразийский журнал технологий и инноваций*, 1(4), 75–84.
9. 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.
10. 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.
11. 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.