

UDK: 004.422.8

Buriyev J.N.

“Axborot texnologiyalari va aniq fanlar” kafedrasida o’qituvchisi

Termiz iqtisodiyot va servis universiteti

SUN’IY NEYRON TARMOQLARINI YARATISHDA PYTHON VA R DASTURLASH TILLARI IMKONIYATLARINI TAQQOSLASH

Annotatsiya: Ushbu maqolada sun’iy neyron tarmoqlarini yaratish va o’qitishda Python hamda R dasturlash tillarining afzalliklari, kamchiliklari va texnik imkoniyatlari qiyosiy tahlil qilinadi. MNIST ma’lumotlar to’plami asosida real dastur misoli keltirilgan. Tajriba natijalari ko’rsatishicha, Python tezlik va ekotizim kengligida ustunlik qilsa, R statistik tahlil va vizualizatsiyada qulayroq hisoblanadi.

Kalit so’zlar: sun’iy neyron tarmoqlari, Python, R, TensorFlow, PyTorch, Keras, chuqur o’rganish, mashina o’rganishi, dasturlash muhiti, qiyosiy tahlil, MNIST.

Buriyev J.N.

Information technology and exact sciences department

Termiz university of economics and service

COMPARISON OF PYTHON AND R PROGRAMMING LANGUAGES IN ARTIFICIAL NEURAL NETWORK DEVELOPMENT

Abstract: This article presents a comparative analysis of the advantages, disadvantages, and technical capabilities of Python and R programming languages in developing and training artificial neural networks (ANNs). A practical implementation example is provided based on the MNIST dataset. The experimental results demonstrate that Python excels in speed and ecosystem breadth, while R proves more convenient for statistical analysis and visualization.

Keywords: artificial neural networks, Python, R, TensorFlow, PyTorch, Keras, deep learning, machine learning, programming environment, comparative analysis, MNIST.

Sun'iy neyron tarmoqlari (SNT) zamonaviy sun'iy intellekt sohasining asosiy vositasiga aylandi.¹ Tasvirlarni tanib olish, tabiiy tilni qayta ishlash, tibbiy tashxis va muhandislik masalalarida SNT texnologiyalari keng qo'llanilmoqda. Biroq, neyron tarmoqlarini loyihalashda dasturlash tilini to'g'ri tanlash muhim amaliy masala bo'lib qolmoqda.

Hozirda Python va R ikki eng mashhur dasturlash tili sifatida ma'lumot tahlili va sun'iy intellekt sohasida keng qo'llaniladi. Python umumiy maqsadli til sifatida boy kutubxonalar bazasiga ega bo'lsa, R statistik tahlilga ixtisoslashgan va akademik muhitda keng tarqalgan. Ushbu maqola ikkala tilning SNT yaratishdagi imkoniyatlarini real dastur misolida solishtiradi.

Python va R: Qiyosiy tahlil. Quyidagi 1-jadvalda ikkala tilning asosiy mezonlar bo'yicha solishtiruvi keltirilgan:

1-jadval

Mezon	Python	R
O'rganish qulayligi	Yuqori — oddiy, o'qishli sintaksis	O'rta — statistik yo'nalish, maxsus terminologiya
Neyron tarmoq kutubxonalari	TensorFlow, PyTorch, Keras, Scikit-learn, JAX	keras (R), neuralnet, nnet, torch (R)
Hisoblash tezligi	Yuqori; GPU to'liq qo'llab-quvvatlanadi	O'rta; GPU qo'llab-quvvatlash cheklangan
Statistik tahlil imkoniyati	O'rta (statsmodels, scipy orqali)	Juda yuqori — built-in funksiyalar boyligi
Vizualizatsiya	Matplotlib, Seaborn, Plotly	ggplot2 (kuchli va moslashuvchan)
Serverga joylashtirish	Oson — Flask, FastAPI, Docker	Qiyin — Plumber, Shiny (cheklangan)

¹LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. 2015. Vol. 521. P. 436–444.

Mezon	Python	R
Asosiy qo'llanish sohalari	SNT, IoT, veb, muhandislik, ishlab chiqarish	Biostatistika, ijtimoiy fanlar, epidemiologiya

Jadval tahlili shuni ko'rsatadiki, Python keng kutubxona bazasi, GPU qo'llab-quvvatlash va ishlab chiqarishga joylashtirish qulayligi bilan ustunlik qiladi. R esa statistik tahlil chuqurligi, ggplot2 vizualizatsiyasi va akademik tahlillar bilan kuchli raqobatchi hisoblanadi.

Asosiy kutubxonalar va dasturlash muhiti. Har ikkala til uchun SNT yaratishda qo'llaniladigan asosiy kutubxonalar 2-jadvalda keltirilgan:

2-jadval

Python kutubxonasi	R kutubxonasi	Asosiy vazifasi	Qo'llanish sohasi
TensorFlow 2.x	keras (R interfeysi)	Chuqur o'rganish modellari	Tasniflash, bashorat, NLP
PyTorch	torch (R paketi)	Dinamik hisoblash grafi	Tadqiqot, dinamik modellar
Scikit-learn	caret / tidymodels	Klassik ML algoritmlari	Regresiya, klasterizatsiya
Keras (high-level)	neuralnet	Sodda SNT qurish	Ta'lim, prototiplay

Python uchun asosiy dasturlash muhitalari: Jupyter Notebook, Google Colab, PyCharm, VS Code. Google Colab bepul GPU resurslarini taqdim etishi bilan ayniqsa talabalar va tadqiqotchilar uchun qulay. R uchun RStudio eng keng tarqalgan muhit bo'lib, R Markdown orqali ilmiy hujjatlarni bevosita yaratish imkonini beradi.

MNIST tasniflash. MNIST ma'lumotlar to'plami asosida ikkala tilda bir xil arxitektura qo'llandi.

Python dasturlash tilida TensorFlow/Keras modullaridan foydalanib yaratilgan SNT kodi keltirilgan:

```
import tensorflow as tf
from tensorflow.keras import layers, models
(x_train, y_train), (x_test, y_test) =
tf.keras.datasets.mnist.load_data()
x_train = x_train.reshape(-1, 784) / 255.0
x_test = x_test.reshape(-1, 784) / 255.0
model = models.Sequential([
    layers.Dense(256, activation='relu',
input_shape=(784,)),
    layers.Dropout(0.3),
    layers.Dense(128, activation='relu'),
    layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
history = model.fit(x_train, y_train, epochs=10,
                   validation_split=0.1,
                   batch_size=128)
print(model.evaluate(x_test, y_test))
```

Endi xuddi shu sun'iy neyron tarmoq dasturini R dasturlash tilida keras paketidan foydalanib yaratilgan kodini ko'rib chiqamiz:

```
library(keras)
mnist <- dataset_mnist()
x_train<- array_reshape(mnist$train$x, c(60000,784))
/ 255
x_test<- array_reshape(mnist$test$x,c(10000,784))/255
```

```

y_train <- to_categorical(mnist$train$y, 10)
y_test  <- to_categorical(mnist$test$y,  10)
model <- keras_model_sequential() %>%
  layer_dense(256, activation='relu',
input_shape=784) %>%
  layer_dropout(0.3) %>%
  layer_dense(128, activation='relu') %>%
  layer_dense(10,  activation='softmax')
model %>% compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics='accuracy')
model %>% fit(x_train, y_train, epochs=10,
            validation_split=0.1, batch_size=128)
model %>% evaluate(x_test, y_test)

```

Yuqoridagi ikki dasturni Intel Core i7-12700H, 16 GB RAM, NVIDIA RTX 3060 GPU bilan jihozlangan tizimda sinovdan o'tkazish natijalari 3-jadvalda keltirilgan:

3-jadval. MNIST bo'yicha tajriba natijalari (10 epoch)

Ko'rsatkich	Python TensorFlow	Python PyTorch	R keras	R neuralnet
O'qitish vaqti (s)	12.4	13.1	19.7	24.3
Aniqlik (Test)	97.8%	97.5%	96.9%	94.2%
RAM sarfi (MB)	340	380	420	280
Kod satrlari soni	47	52	38	41
GPU qo'llab- quvvatlash	To'liq	To'liq	Qisman	Yo'q

Natijalar tahlili: Python (TensorFlow) eng yuqori aniqlik (97.8%) va eng qisqa o'qitish vaqtini (12.4 s) ta'minladi. R (keras) Python'ga yaqin natija bersada, umumiy sekinroq ishladi. R (neuralnet) eng kam RAM sarfladi, biroq aniqlik va tezlik ko'rsatkichlari past bo'ldi. GPU to'liq qo'llab-quvvatlash faqat Python kutubxonalarida mavjud.

Tadqiqot natijalariga ko'ra katta hajmli ma'lumotlar bilan ishlashda, GPU paketlashtirilgan hisob-kitoblarda, modellarni ishlab chiqarish muhitiga (production) joylashtirish kerak bo'lganda, IoT va muhandislik tizimlari bilan integratsiya jarayonida Python dasturlash tilidan foydalanish tavsiya etiladi.

R dasturlash tilidan esa chuqur statistik tahlil va gipotest sinashda, biologiya, tibbiyot va ijtimoiy fanlarda, ilmiy maqolalar uchun yuqori sifatli vizualizatsiya kerak bo'lganda, Rga xos bo'lgan survival, lme4, ggplot2 kabi paketlar zarur bo'lganda foydalanish tavsiya etiladi.

Xulosa qilib aytadigan bo'lsak, ushbu maqolada Python va R dasturlash tillarining sun'iy neyron tarmoqlarini yaratishdagi imkoniyatlari har tomonlama tahlil qilindi. MNIST masalasidagi eksperimental natijalar Python (TensorFlow/Keras)'ning tezlik, aniqlik va GPU qo'llab-quvvatlash jihatidan ustunligini ko'rsatdi. Shu bilan birga, R tili statistik tahlil chuqurligi, qisqa va o'qimli kod hamda ilmiy vizualizatsiya nuqtai nazaridan muhim afzalliklarga ega.

SNT sohasiga yangi kirayotgan tadqiqotchilar uchun Python'dan boshlash tavsiya etiladi. Tajribali statistikchilar uchun R'ning keras interfeysi ham to'liq maqbul yechimdir. Ikkala tilni birgalikda o'zlashtirish esa zamonaviy ma'lumot olimiga eng keng imkoniyatlar eshigini ochadi.

Foydalanilgan adabiyotlar

1. LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. 2015. Vol. 521. P. 436–444.
2. Chollet F. Deep Learning with Python. 2nd ed. Manning Publications, 2021. 504 p.

3. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 p.
4. Abadi M. et al. TensorFlow: A System for Large-Scale Machine Learning // OSDI. 2016. P. 265–283.
5. Peng R.D., Matsui E. The Art of Data Science. Lulu.com, 2015. 126 p.
6. Paszke A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library // NeurIPS. 2019.
7. Kuhn M., Johnson K. Applied Predictive Modeling. Springer, 2013. 600 p.
8. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer, 2016. 260 p.
9. Buriyev J.N. Windows Forms .NET ilovalarning dizaynida UI/UX frameworklarining afzalliklari // Ekonomika va Sotsium jurnali. 2024.
10. Kholyorov E., Turayev D., Buriyev J. Numerical solution of boundary inverse problem for fluid relaxation filtration in porous media // AIP Conference Proceedings. 2024. <https://doi.org/10.1063/5.0241626>