

**NEYRON TO'RLARI VA KLASTERLI TAHLILNI ALGORITMLASH
VA MODELLASHTIRISH**

Xoliqova Manzura Qoyirovna

Toshkent davlat agrar universiteti,

“Oliy matematika” kafedrası assistenti

tel.: +998934758834, manzuraxoliqova37@gmail.com ,

https://orcid.org/0009-0004-5554-0677

Xakimova Indira Kurmangaliyevna

Toshkent davlat transport universiteti, “Informatik

ava kompyuter grafikasi” kafedrası assistenti,

tel.: +998935990703, xakimova.indira64@gmail.com ,

https://orcid.org/0009-0002-2905-1680

Anotatsiya: Ushbu maqolada neyron tarmoqlar va klasterli tahlilning asosiy tushunchalari, ularning algoritmlari va modellashtirish jarayoni yoritiladi. Neyron tarmoqlarning ishlash tamoyillari, K-means klasterlash algoritmi va ularni dasturiy jihatdan amalga oshirish usullari tushuntiriladi. Shuningdek, neyron tarmoqlar va klasterli tahlilning turli sohalarda qo‘llanilishi ham ko‘rib chiqiladi.

Kalit so‘zlar: neyron tarmoqlar, model, algoritim, qatlam, klasterli tahlil, K-means, sun‘iy intellekt, modellashtirish, algoritmlash

Abstract: This article explores the fundamental concepts of neural networks and cluster analysis, their algorithms, and the modeling process. It explains the operating principles of neural networks, the K-means clustering algorithm, and methods for their software implementation. Furthermore, the application of neural networks and cluster analysis in various fields is examined.

Keywords: neural networks, model, algorithm, layer, cluster analysis, K-means, artificial intelligence, modeling, algorithmization.

Аннотация: В данной статье освещаются основные понятия нейронных сетей и кластерного анализа, их алгоритмы и процесс моделирования. Объясняются принципы работы нейронных сетей, алгоритм кластеризации K-means и методы их программной реализации. Также рассматривается применение нейронных сетей и кластерного анализа в различных сферах.

Ключевые слова: нейронные сети, модель, алгоритм, слой, кластерный анализ, K-means, искусственный интеллект, моделирование, алгоритмизация.

Zamonaviy sun‘iy intellekt texnologiyalari orasida neyron tarmoqlar va klasterli tahlil muhim o‘rin tutadi. Neyron tarmoqlar katta hajmdagi ma’lumotlarni qayta ishlash va o‘rganish uchun ishlatilsa, klasterli tahlil ma’lumotlarni guruhlash va

ularning xususiyatlarini aniqlash uchun ishlatiladi. Ushbu maqolada neyron tarmoqlar va klasterli tahlilning algoritmlari, dasturlash asoslari va modellashtirish jarayoni yoritiladi.

Neyron tarmoqlar va ularning ishlash tamoyillari

Neyron tarmoqlar – sun’iy intellektning asosiy vositalaridan biri bo‘lib, ular inson miyasining ishlash prinsiplariga asoslanadi. Ular quyidagi asosiy komponentlardan iborat:

- **Kirish qatlami** – dastlabki ma’lumotlar qabul qilinadigan qatlam.
- **Yashirin qatlamlar** – neyronlarning bir-biri bilan bog‘langan qismi, u yerda ma’lumotlar qayta ishlanadi.
- **Chiqish qatlami** – yakuniy natijalarni chiqaruvchi qatlam.

Neyron tarmoqlar **faollashtirish funksiyalari** (ReLU, Sigmoid, Softmax) va **orqaga tarqalish algoritmi** orqali o‘qitiladi. Bu jarayon modelni aniqroq va samarali qilishga yordam beradi.

Klasterli tahlil va K-means algoritmi

Klasterli tahlil ma’lumotlarni o‘xshashliklariga qarab guruhlash jarayonidir. Eng mashhur algoritmlardan biri **K-means** bo‘lib, u quyidagi bosqichlardan iborat:

1. **Boshlang‘ich markazlarni tanlash** – K ta klasterlarni boshlang‘ich nuqtalari tanlanadi.
2. **Har bir nuqtani eng yaqin klasterga tayinlash** – Ma’lumotlar ushbu markazlarga asoslangan holda guruhlanadi.
3. **Klaster markazlarini yangilash** – Har bir klasterning o‘rtacha qiymati hisoblanib, yangi markazlar aniqlanadi.
4. **Jarayonni takrorlash** – Klaster markazlari barqaror bo‘lguncha bosqichlar davom ettiriladi.

Neyron tarmoqlar va klasterlashni modellashtirish

Ushbu modellashtirish jarayoni quyidagicha amalga oshiriladi:

1. **Ma’lumotlarni tayyorlash** – Tasodifiy ma’lumotlar hosil qilinadi va standartlashtiriladi.
2. **K-means algoritmi orqali klasterlash** – Ma’lumotlar 3 ta klasterga ajratiladi.
3. **Neyron tarmoq yaratish** – 3 qavatli neyron tarmoq tuziladi va K-means natijalari asosida o‘qitiladi.
4. **Modelni baholash** – Modelning aniqligi hisoblanadi va natijalar tahlil qilinadi.

Dastur quyidagi barcha funksiyalarni o‘z ichiga oladi:

- a) K-means klasterlash
- b) Neyron tarmoqni to‘liq modellashtirish
- c) Natijalarni vizualizatsiya qilish
- d) Grafik interfeys (GUI)

- e) Murakkab tarmoq arxitekturasi
 f) Ma'lumotlarni yuklash va saqlash funksiyasi

Bu dastur kodi quyidagicha bo'ladi:

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout,
QPushButton, QLabel, QFileDialog
from PyQt5.QtCore import Qt
import matplotlib.pyplot as plt
import pickle

# Sun'iy ma'lumotlar yaratish
def generate_data ():
    np.random.seed (42)
    data = np.random.rand (500, 2)
    return data

# K-means algoritmi yordamida klasterlash
def kmeans_clustering(data, n_clusters=3):
    scaler = StandardScaler ()
    data_scaled = scaler.fit_transform(data)
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    kmeans.fit(data_scaled)
    labels = kmeans.labels_
    return labels, data_scaled

# Neyron tarmoq modelini yaratish
def create_model(input_dim=2, n_clusters=3):
    model = keras.Sequential([
        keras.layers.Dense(8, activation='relu', input_shape=(input_dim,)),
        keras.layers.Dense(8, activation='relu'),
        keras.layers.Dense(n_clusters, activation='softmax')])
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
    return model
```

```
# Modelni o'rgatish
def train_model(model, data, labels):
    model.fit(data, labels, epochs=50, verbose=1)
    return model

# Modelni baholash
def evaluate_model(model, data, labels):
    loss, accuracy = model.evaluate(data, labels)
    return accuracy

# Natijalarni vizualizatsiya qilish
def plot_clusters(data, labels):
    plt.scatter(data[:,0], data[:,1], c=labels, cmap='viridis')
    plt.title('K-means Klasterlash')
    plt.show()

# Ma'lumotlarni saqlash va yuklash
def save_model(model, filename="model.h5"):
    model.save(filename)

def load_model(filename="model.h5"):
    return keras.models.load_model(filename)

def save_data(data, filename="data.pkl"):
    with open(filename, 'wb') as f:
        pickle.dump(data, f)

def load_data(filename="data.pkl"):
    with open(filename, 'rb') as f:
        return pickle.load(f)

# GUI uchun PyQt5
class App(QWidget):
    def __init__(self):
        super().init()
        self.setWindowTitle('Neyron Tarmoq va Klasterlash')
        self.setGeometry(100, 100, 300, 200)
        self.layout = QVBoxLayout()
```

```

self.label = Q Label ('Boshqaruv paneli', self)
self.layout.addWidget(self.label)

self.load_button = QPushButton('Ma'lumotni yuklash', self)
self.layout.addWidget(self.load_button)
self.load_button.clicked.connect(self.load_data_handler)

self.train_button = QPushButton('Modelni o'rgatish', self)
self.layout.addWidget(self.train_button)
self.train_button.clicked.connect(self.train_model_handler)

self.save_button = QPushButton ('Modelni saqlash', self)
self.layout.addWidget(self.save_button)
self.save_button.clicked.connect(self.save_model_handler)

self.evaluate_button = QPushButton('Modelni baholash', self)
self.layout.addWidget(self.evaluate_button)
self.evaluate_button.clicked.connect(self.evaluate_model_handler)

self.plot_button = QPushButton ('Natijalarni vizualizatsiya qilish', self)
self.layout.addWidget(self.plot_button)
self.plot_button.clicked.connect(self.plot_handler)

self.setLayout(self.layout)

def load_data_handler(self):
    file_name, _ = QFileDialog.getOpenFileName(self, "Ma'lumotni yuklash",
    "", "All Files (*)")
    if file_name:
        data = load_data(file_name)
        self.data = data
        self.label.setText ('Ma'lumotlar yuklandi.')

def train_model_handler(self):
    if hasattr (self, 'data'):
        labels, data_scaled = kmeans_clustering(self.data)
        self.model = create_model ()
        self.model = train_model (self.model, data_scaled, labels)

```

```

self.label.setText ('Model o'rgatildi.')
else:
self.label.setText ('Iltimos, ma'lumotlarni yuklang.')

def save_model_handler(self):
if hasattr(self, 'model'):
file_name, _ = QFileDialog.getSaveFileName(self, "Modelni saqlash",
"", "H5 Files (*.h5)")
if file_name:
save_model (self.model, file_name)
self.label.setText ('Model saqlandi.')
else:
self.label.setText ('Iltimos, modelni o'rgating.')

def evaluate_model_handler(self):
if hasattr(self, 'model'):
accuracy = evaluate_model(self.model, self.data, self.labels)
self.label.setText (f'Model aniqligi: {accuracy * 100:.2f} %')
else:
self.label.setText ('Iltimos, modelni o'rgating.')

def plot_handler(self):
if hasattr(self, 'data'):
labels, data_scaled = kmeans_clustering(self.data)
plot_clusters (data_scaled, labels)
else:
self.label.setText ('Iltimos, ma'lumotlarni yuklang.')

# Dastur ishga tushishi
def main ():
app = QApplication ([])
window = App ()
window.show()
app.exec()
if __name__ == '__main__':
main ()

```

Xulosa

Neyron tarmoqlar va klasterli tahlil turli sohalarda qo'llaniladigan kuchli texnologiyalardir. Ushbu usullar katta hajmdagi ma'lumotlarni samarali guruhlash va

bashorat qilish imkonini beradi. Kelajakda ushbu texnologiyalar yanada rivojlanib, yanada samarali modellar yaratishga yordam beradi. Neyron tarmoqlar va klasterli tahlil zamonaviy sun'iy intellekt va ma'lumotlarni qayta ishlash sohalarida muhim o'rin tutadi. Ushbu maqolada ushbu ikki metodologiyaning o'zaro uyg'unligi, ularning murakkab ma'lumotlar to'plamlarini tahlil qilishdagi samaradorligi tahlil qilindi. Neyron tarmoqlar yordamida yuqori o'lchamli va notekis ma'lumotlar strukturalashtirildi hamda ulardan yashirin xususiyatlar ajratib olindi. Shundan so'ng, klasterlash algoritmlari (masalan, K-means, DBSCAN, yoki Hierarchical Clustering) vositasida ma'lumotlar tabiiy guruhlariga ajratildi.

Tadqiqotlar shuni ko'rsatdiki, neyron tarmoqlar tomonidan olingan xususiyatlar asosida klasterlash natijalari sezilarli darajada aniqroq va ma'noliroq bo'ladi. Ayniqsa, avtoenkoderlar yoki chuqur o'rganishga asoslangan arxitekturalar klasterlash samaradorligini oshiradi. Bunday yondashuv tibbiyot, moliya, kibernetika, kiberxavfsizlik, hamda tavsiyaviy tizimlarda keng amaliy ahamiyatga ega bo'lishi mumkin.

Kelgusidagi tadqiqotlarda neyron tarmoqlar asosidagi o'z-o'zini o'rganuvchi (self-supervised learning) yondashuvlar va grafik neyron tarmoqlari bilan birgalikda klasterli tahlilni integratsiyalash orqali yanada chuqurroq tahlil metodologiyalarini ishlab chiqish istiqbollari mavjud.

Foydalanilgan adabiyotlar

1. Thematic Journal of Applied Sciences (ISSN 2277-3037) Volume 6 Issue1
<https://doi.org/10.5281/zenodo.6382042>, B. 70-73
2. Science and innovation international scientific journal volume 1 ISSUE 6 UIF-2022: 8.2 | ISSN: 2181-3337, B. 282-286
3. N.Noraliyev., Iqtisodiyotda axborot kommunikatsion texnologiyalar, 2022 y.
4. Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning, 2016
5. Michael Nielsen – Neural Networks and Deep Learning,
<http://neuralnetworksanddeeplearning.com>2015y.
6. Francois Chollet – Deep Learning with Python, 2021y.
7. Simon Haykin – Neural Networks and Learning Machines, 2008 y.
8. Charu C. Aggarwal – Neural Networks and Deep Learning: A Textbook, 2018y.
9. N.Noraliyev., S. Rasulov. Axborot kommunikatsion texnologiyalar, Darslik, -T. "Iqtisod-moliya", -2020 y.