Жарқын Ақтілек

10. Автокодерлерді медициналық бейнелерді өңдеу немесе ауруларды диогностикалауда қолдану

import tensorflow as tf

from tensorflow.keras import layers, models

import numpy as np

import matplotlib.pyplot as plt

# MNIST-пен мысал (медициналық емес, бірақ идеяны түсінуге жақсы)

(x\_train, \_), (x\_test, \_) = tf.keras.datasets.mnist.load\_data()

x\_train = x\_train.astype('float32') / 255.

x\_test = x\_test.astype('float32') / 255.

x\_train = np.reshape(x\_train, (len(x\_train), 28, 28, 1))

x\_test = np.reshape(x\_test, (len(x\_test), 28, 28, 1))

# Автокодер моделі

input\_img = layers.Input(shape=(28, 28, 1))

x = layers.Conv2D(16, (3, 3), activation='relu', padding='same')(input\_img)

x = layers.MaxPooling2D((2, 2), padding='same')(x)

x = layers.Conv2D(16, (3, 3), activation='relu', padding='same')(x)

x = layers.UpSampling2D((2, 2))(x)

decoded = layers.Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

autoencoder = models.Model(input\_img, decoded)

autoencoder.compile(optimizer='adam', loss='binary\_crossentropy')

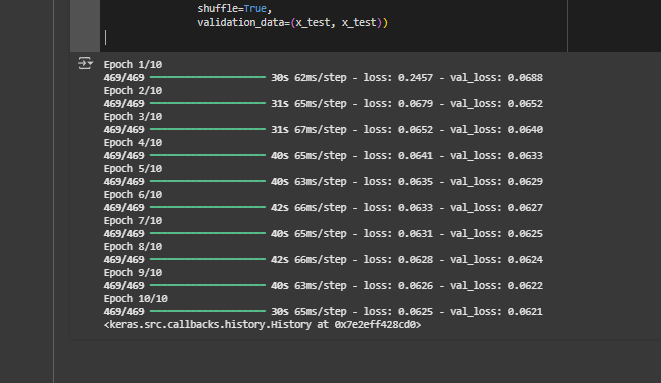
autoencoder.fit(x\_train, x\_train,

                epochs=10,

                batch\_size=128,

                shuffle=True,

                validation\_data=(x\_test, x\_test))



11. Автокодерлерді шуларды жою үшінқолданудың мысалын келтіріңіз. Бұл жағдайдаавтокодердің құрылымы

import numpy as np

import matplotlib.pyplot as plt

from tensorflow.keras.datasets import mnist

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Input, Dense

from tensorflow.keras import layers

# Деректерді жүктеу

(x\_train, \_), (x\_test, \_) = mnist.load\_data()

# Деректерді масштабтау (0-1)

x\_train = x\_train.astype('float32') / 255.

x\_test = x\_test.astype('float32') / 255.

x\_train = x\_train.reshape((len(x\_train), 28 \* 28))

x\_test = x\_test.reshape((len(x\_test), 28 \* 28))

# Шум қосу

noise\_factor = 0.5

x\_train\_noisy = x\_train + noise\_factor \* np.random.normal(loc=0.0, scale=1.0, size=x\_train.shape)

x\_test\_noisy = x\_test + noise\_factor \* np.random.normal(loc=0.0, scale=1.0, size=x\_test.shape)

x\_train\_noisy = np.clip(x\_train\_noisy, 0., 1.)

x\_test\_noisy = np.clip(x\_test\_noisy, 0., 1.)

# Автокодер архитектурасы

input\_img = Input(shape=(784,))

encoded = Dense(128, activation='relu')(input\_img)

encoded = Dense(64, activation='relu')(encoded)

decoded = Dense(128, activation='relu')(encoded)

decoded = Dense(784, activation='sigmoid')(decoded)

autoencoder = Model(input\_img, decoded)

autoencoder.compile(optimizer='adam', loss='binary\_crossentropy')

# Модельді оқыту

autoencoder.fit(x\_train\_noisy, x\_train,

                epochs=10,

                batch\_size=256,

                shuffle=True,

                validation\_data=(x\_test\_noisy, x\_test))

# Нәтижені визуализациялау

decoded\_imgs = autoencoder.predict(x\_test\_noisy)

n = 10

plt.figure(figsize=(20, 4))

for i in range(n):

    # Шулы кескін

    ax = plt.subplot(3, n, i + 1)

    plt.imshow(x\_test\_noisy[i].reshape(28, 28), cmap="gray")

    plt.axis("off")

    # Тазаланған кескін

    ax = plt.subplot(3, n, i + 1 + n)

    plt.imshow(decoded\_imgs[i].reshape(28, 28), cmap="gray")

    plt.axis("off")

    # Бастапқы (таза) кескін

    ax = plt.subplot(3, n, i + 1 + 2\*n)

    plt.imshow(x\_test[i].reshape(28, 28), cmap="gray")

    plt.axis("off")

plt.show()



12. Автокодерлерді ансамбльдік әдістерменбіріктірудің тиімділігі туралы ойыңыздыжазыңыз. Бұл әдіс классификация немесе егрессия мәселелері

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

from keras.models import Model

from keras.layers import Input, Dense

from keras.optimizers import Adam

# Жасанды мәліметтер жасау (1000 үлгі, 20 белгі)

X = np.random.rand(1000, 20)

y = np.random.randint(0, 2, size=(1000,))  # 0 немесе 1 класстар

# Мәліметтерді train және test-ке бөлу

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# --- Автокодер ---

input\_dim = X\_train.shape[1]

input\_layer = Input(shape=(input\_dim,))

encoded = Dense(10, activation='relu')(input\_layer)

decoded = Dense(input\_dim, activation='sigmoid')(encoded)

autoencoder = Model(inputs=input\_layer, outputs=decoded)

autoencoder.compile(optimizer=Adam(), loss='mse')

autoencoder.fit(X\_train, X\_train, epochs=20, batch\_size=32, verbose=0)

# Енкодер арқылы жаңа белгілерді алу

encoder = Model(inputs=input\_layer, outputs=encoded)

X\_train\_encoded = encoder.predict(X\_train)

X\_test\_encoded = encoder.predict(X\_test)

# --- Ансамбльдік әдіс (Random Forest) ---

rf = RandomForestClassifier()

rf.fit(X\_train\_encoded, y\_train)

# Болжау және нәтижені бағалау

y\_pred = rf.predict(X\_test\_encoded)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Модель дәлдігі (accuracy):", accuracy)

