

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №4
По дисциплине: «ИАД»
Тема: «**Предобучение нейронных сетей с использованием RBM**»

Выполнил:
Студент 4 курса
Группы ИИ-23
Романюк А. П.
Проверила:
Андренко К.В.

Брест 2025

Цель: научиться осуществлять предобучение нейронных сетей с помощью RBM

Общее задание

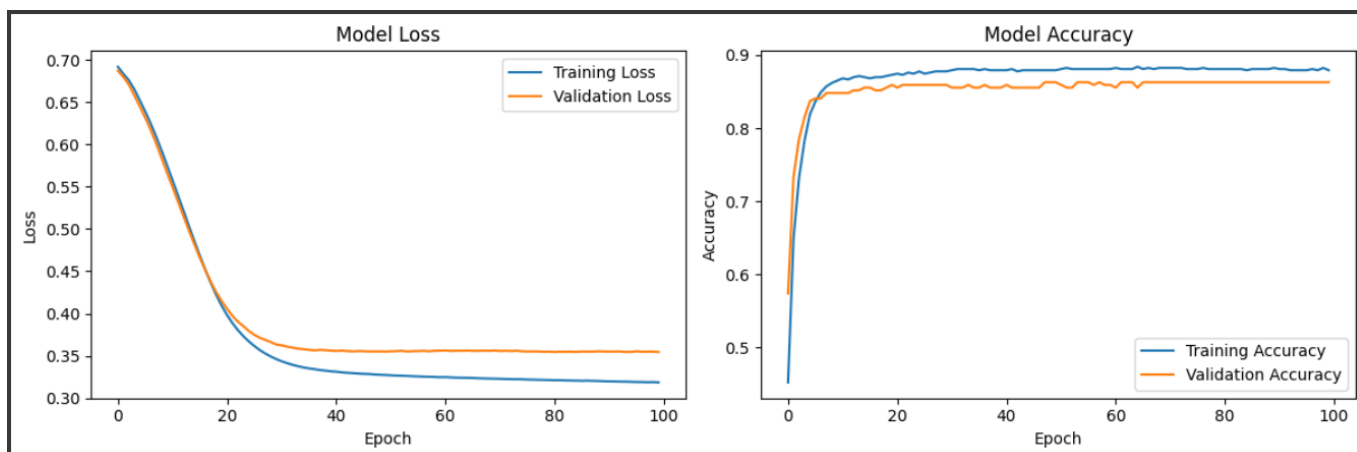
1. Взять за основу нейронную сеть из лабораторной работы №3. Выполнить обучение с предобучением, используя стек ограниченных машин Больцмана (RBM – Restricted Boltzmann Machine), алгоритм которого изложен в лекции. Условие останова (например, по количеству эпох) при обучении отдельных слоев как RBM выбрать самостоятельно.
2. Сравнить результаты, полученные при
 - обучении без предобучения (ЛР 3);
 - обучении с предобучением, используя автоэнкодерный подход (ЛР3);
 - обучении с предобучением, используя RBM.
3. Обучить модели на данных из ЛР 2, сравнить результаты по схеме из пункта 2;
4. Сделать выводы, оформить отчет по выполненной работе, загрузить исходный код и отчет в соответствующий репозиторий на github.

Задание по вариантам

9	https://archive.ics.uci.edu/dataset/850/raisin	классификация	Class
---	---	---------------	-------

Код программы:

Модель без предобучения:



```

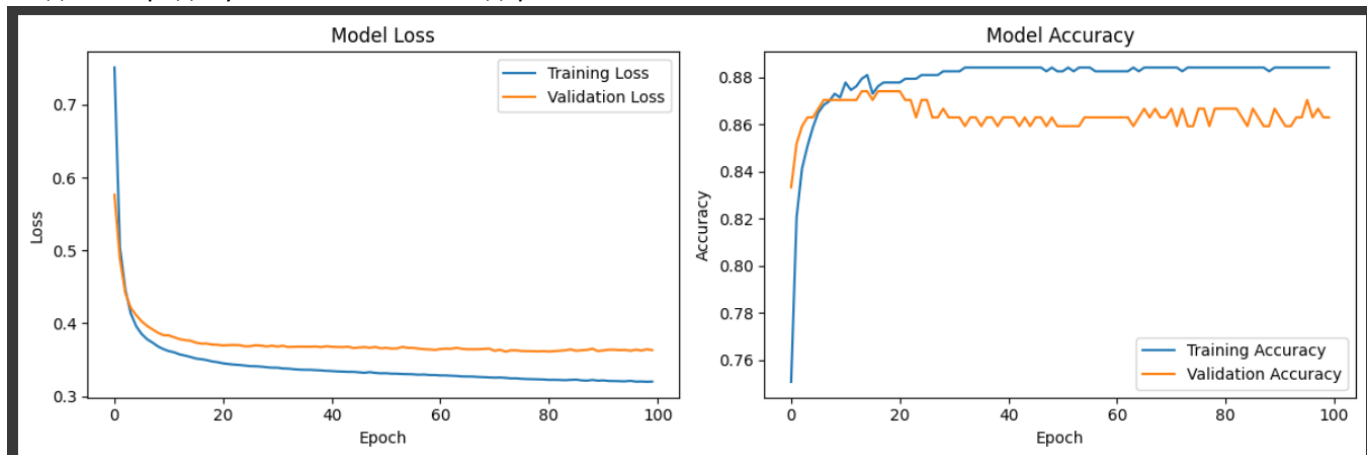
Classification Report:
              precision    recall  f1-score   support

     0       0.85         0.87         0.86         129
     1       0.88         0.86         0.87         141

   accuracy              0.86         270
  macro avg              0.86         0.86         0.86         270
 weighted avg              0.86         0.86         0.86         270

Confusion Matrix:
[[112  17]
 [ 20 121]]
  
```

Модель с предобучением на автоэнеодере:



Classification Report (Pretrained):

	precision	recall	f1-score	support
0	0.85	0.87	0.86	129
1	0.88	0.86	0.87	141
accuracy			0.86	270
macro avg	0.86	0.86	0.86	270
weighted avg	0.86	0.86	0.86	270

Confusion Matrix (Pretrained):

```
[[112  17]
 [ 20 121]]
```

Предобучение с машинами Больцмана:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
import matplotlib.pyplot as plt

data = pd.read_excel("Raisin_Dataset.xlsx")

label_encoder = LabelEncoder()
data['Class'] = label_encoder.fit_transform(data['Class'])

X = data.drop('Class', axis=1)
y = data['Class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```

from sklearn.neural_network import BernoulliRBM
layer_sizes = [64, 32, 16, 8]
rbm_weights = []
rbm_biases = []

input_data = X_train_scaled.copy()

for size in layer_sizes:
    rbm = BernoulliRBM(n_components=size, learning_rate=0.01, batch_size=16,
n_iter=20, random_state=42)
    rbm.fit(input_data)
    rbm_weights.append(rbm.components_.T) # shape: (input_dim,
hidden_dim)
    rbm_biases.append(rbm.intercept_hidden_)
    input_data = rbm.transform(input_data)

from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam

model = Sequential()
model.add(Dense(64, input_shape=(X_train.shape[1],), activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(2, activation='softmax'))

model.compile(optimizer=Adam(learning_rate=0.0005),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

history = model.fit(X_train_scaled, y_train,
                    validation_data=(X_test_scaled, y_test),
                    epochs=100,
                    batch_size=16)

plt.figure(figsize=(12, 4))

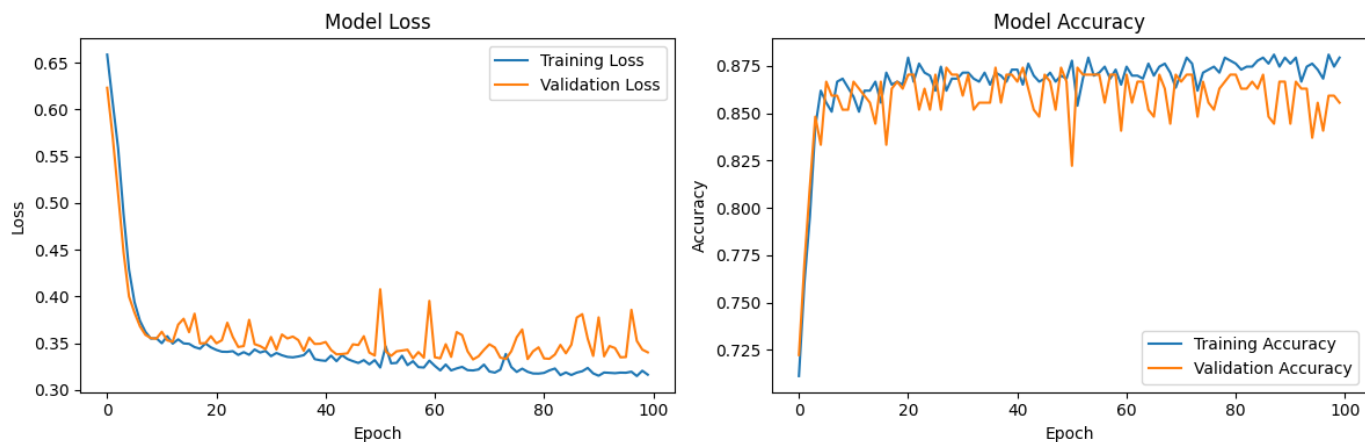
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epoch')

```

```
plt.ylabel('Accuracy')
plt.legend()
```

```
plt.tight_layout()
plt.show()
```



```
from sklearn.metrics import classification_report, confusion_matrix
import numpy as np
```

```
y_pred = np.argmax(model.predict(X_test), axis=-1)
```

```
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

     0       0.85         0.84         0.85         129
     1       0.86         0.87         0.86         141

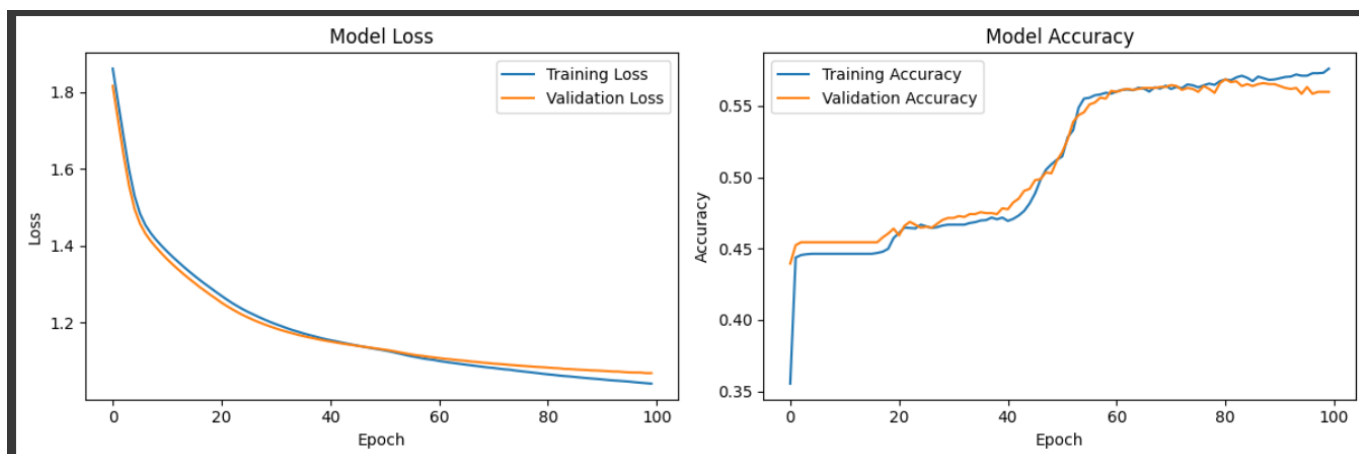
   accuracy              0.86         270
  macro avg       0.86         0.86         0.86         270
 weighted avg       0.86         0.86         0.86         270

Confusion Matrix:
[[109  20]
 [ 19 122]]
```

Для задач с подобными характеристиками (небольшой объем данных, четкая разделимость классов) сложные методы предобучения не являются необходимыми. Стандартная архитектура нейронной сети показывает максимальную эффективность "из коробки", обеспечивая при этом более простое и быстрое решение. Методы предобучения, такие как автоэнкодер и RBM, раскрывают свой потенциал в задачах со более сложной структурой данных, большей размерностью или при недостатке размеченных примеров для обучения.

Данные для датасета winequality-white:

С нуля:



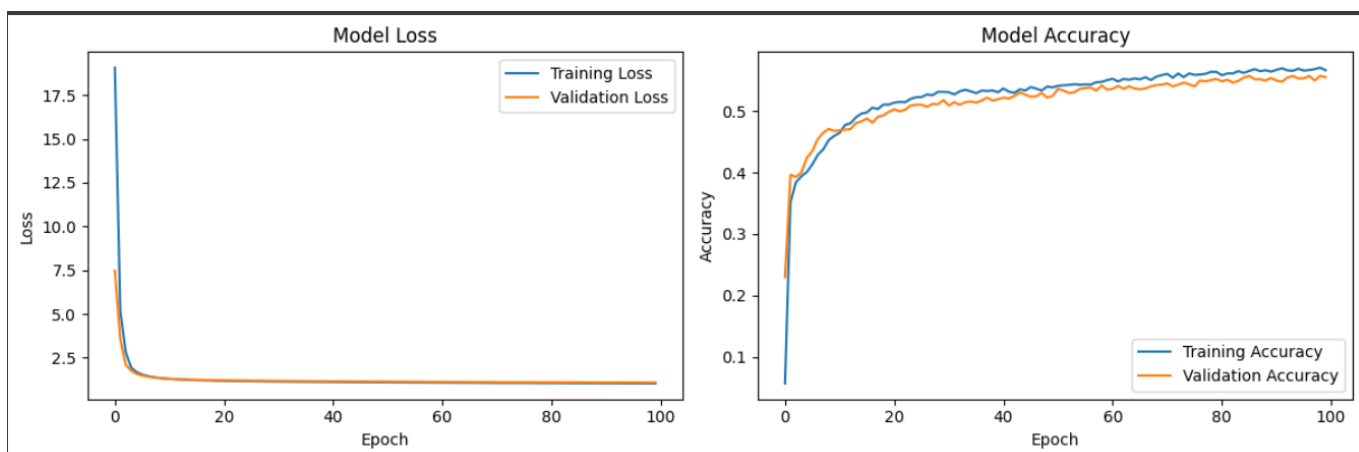
Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	7
1	0.00	0.00	0.00	40
2	0.59	0.65	0.62	426
3	0.55	0.69	0.61	668
4	0.52	0.29	0.37	280
5	0.00	0.00	0.00	49
accuracy			0.56	1470
macro avg	0.28	0.27	0.27	1470
weighted avg	0.52	0.56	0.53	1470

Confusion Matrix:

```
[[ 0  0  2  5  0  0]
 [ 0  0 24 16  0  0]
 [ 0  0 279 142  5  0]
 [ 0  0 153 464 51  0]
 [ 0  0 10 190 80  0]
 [ 0  0  1 29 19  0]]
```

Предтренированная на автоэнкодере:



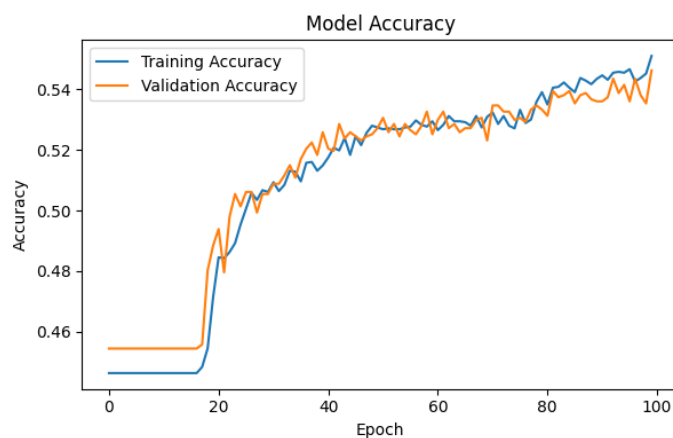
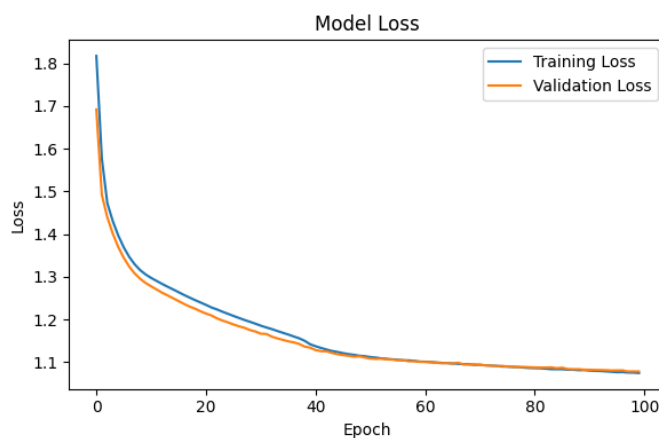
Classification Report (Pretrained):

	precision	recall	f1-score	support
0	0.00	0.00	0.00	7
1	0.00	0.00	0.00	40
2	0.61	0.57	0.59	426
3	0.54	0.74	0.63	668
4	0.50	0.28	0.36	280
5	0.00	0.00	0.00	49
accuracy			0.56	1470
macro avg	0.28	0.27	0.26	1470
weighted avg	0.52	0.56	0.52	1470

Confusion Matrix (Pretrained):

```
[[ 0  0  2  5  0  0]
 [ 0  0 23 17  0  0]
 [ 0  3 241 175  7  0]
 [ 0  0 119 497 52  0]
 [ 0  0  7 194 79  0]
 [ 0  0  1 29 19  0]]
```

Предтренированная на RBM:



```

Classification Report:
              precision    recall  f1-score   support

     0       0.50      0.14      0.22         7
     1       0.32      0.15      0.20        40
     2       0.59      0.62      0.61       426
     3       0.56      0.72      0.63       668
     4       0.57      0.28      0.37       280
     5       0.00      0.00      0.00         49

 accuracy      0.57      1470
 macro avg      0.42      0.32      0.34      1470
weighted avg      0.54      0.57      0.54      1470

Confusion Matrix:
[[ 1  0  2  4  0  0]
 [ 1  6 23 10  0  0]
 [ 0 10 266 147  3  0]
 [ 0  2 149 480 37  0]
 [ 0  1  9 192 78  0]
 [ 0  0  1 29 19  0]]

```

Метод предобучения на основе Restricted Boltzmann Machines (RBM) демонстрирует преимущество для решения задачи классификации с дисбалансом классов, обеспечивая более сбалансированное качество распознавания среди всех категорий.

Вывод: научился осуществлять предобучение нейронных сетей с помощью автоэнкодерного подхода