

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

Лабораторная работа №2  
По дисциплине «ОИвИС»  
Тема: “Конструирование моделей на базе предобученных  
нейронных сетей”

Выполнил:  
Студент 4 курса  
Группы ИИ-24  
Рекун Р.А.  
Проверила:  
Андренко К.В.

Брест 2025

**Цель:** научиться конструировать нейросетевые классификаторы и выполнять их обучение на известных выборках компьютерного зрения.

## Вариант 2.

## Выборка: Fashion-MNIST

Размер исходного изображения: 28\*28

## Предобученная архитектура: ResNet18

Оптимизатор: SGD

1. Для заданной выборки и архитектуры предобученной нейронной организовать процесс обучения НС, предварительно изменив структуру слоев, в соответствии с предложенной выборкой. Использовать тот же оптимизатор, что и в ЛР №1. Построить график изменения ошибки и оценить эффективность обучения на тестовой выборке;
2. Сравнить полученные результаты с результатами, полученными на кастомных архитектурах из ЛР №1;
3. Ознакомиться с state-of-the-art результатами для предлагаемых выборок (по материалам в сети Интернет). Сделать выводы о результатах обучения НС из п. 1 и 2;
4. Реализовать визуализацию работы предобученной СНС и кастомной (из ЛР 1). Визуализация осуществляется посредством выбора и подачи на сеть произвольного изображения (например, из сети Интернет) с отображением результата классификации;
5. Оформить отчет по выполненной работе, залить исходный код и отчет в соответствующий репозиторий на github.

## 1. Результат работы программы:

```
PS C:\Users\ismem\OneDrive\labs> & "C:/Program Files/Python311/python.exe" c:\Acursissem\OVMK\labs/lab2/lab2.py
Using device: cpu
Загрузка Fashion-MNIST dataset...
Размер обучающей выборки: 60000
Размер тестовой выборки: 10000
Инициализация упрощенной модели...
Модель загружена на сри
Количество параметров: 11,243,466

Начало обучения...

=====
Epoch 1/5 [Train]: 100% | ████████████████████████████████████████████████████████████ | 938/938 [34:24<00:00, 2.20s/it, Loss=0.4509, Acc=88.56%]
Epoch 1/5 [Test]: 100% | ████████████████████████████████████████████████████████████ | 157/157 [01:40<00:00, 1.56it/s, Acc=92.25%]
Epoch [ 1/5 ] Train Loss: 0.3305 Train Acc: 88.56% Test Acc: 92.25%
Epoch 2/5 [Train]: 100% | ████████████████████████████████████████████████████████████ | 938/938 [32:16<00:00, 2.06s/it, Loss=0.2783, Acc=93.18%]
Epoch 2/5 [Test]: 100% | ████████████████████████████████████████████████████████████ | 157/157 [01:44<00:00, 1.51it/s, Acc=92.76%]
Epoch [ 2/5 ] Train Loss: 0.1938 Train Acc: 93.18% Test Acc: 92.76%
Epoch 3/5 [Train]: 100% | ████████████████████████████████████████████████████████████ | 938/938 [28:58<00:00, 1.85s/it, Loss=0.2466, Acc=94.15%]
Epoch 3/5 [Test]: 100% | ████████████████████████████████████████████████████████████ | 157/157 [01:41<00:00, 1.55it/s, Acc=92.88%]
Epoch [ 3/5 ] Train Loss: 0.1658 Train Acc: 94.15% Test Acc: 92.88%
Epoch 4/5 [Train]: 100% | ████████████████████████████████████████████████████████████ | 938/938 [2:04:19<00:00, 7.95s/it, Loss=0.0520, Acc=94.97%]
Epoch 4/5 [Test]: 100% | ████████████████████████████████████████████████████████████ | 157/157 [01:38<00:00, 1.60it/s, Acc=92.91%]
Epoch [ 4/5 ] Train Loss: 0.1419 Train Acc: 94.97% Test Acc: 92.91%
Epoch 5/5 [Train]: 100% | ████████████████████████████████████████████████████████████ | 938/938 [31:00<00:00, 1.98s/it, Loss=0.1890, Acc=95.50%]
Epoch 5/5 [Test]: 100% | ████████████████████████████████████████████████████████████ | 157/157 [02:08<00:00, 1.22it/s, Acc=93.69%]
Epoch [ 5/5 ] Train Loss: 0.1284 Train Acc: 95.50% Test Acc: 93.69%

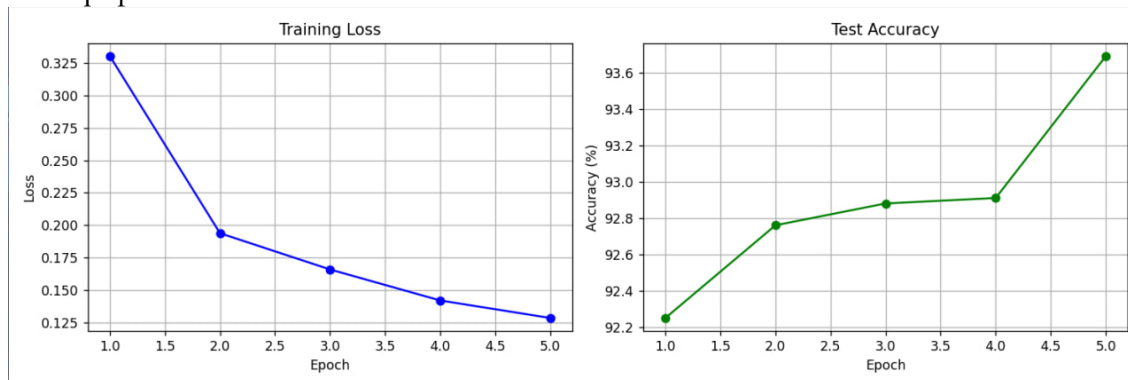
=====
Обучение завершено!
Final Evaluation: 100% | ████████████████████████████████████████████████████████████ | 157/157 [01:44<00:00, 1.51it/s]

Финальная точность на тестовой выборке: 93.69%

Визуализация предсказаний:
Примеры предсказаний: 8/8 правильных
```

Предобученная ResNet18 (упрощенная): 93.69%  
 Кастомная CNN из ЛР1: 92.66%  
 State-of-the-art: ~96.7%

График изменения ошибок:



## 2. Сравнения

Предобученная модель ResNet18 продемонстрировала высокую эффективность уже на первой эпохе обучения, достигнув точности 92.25%, что сопоставимо с результатом кастомной CNN из лабораторной работы №1 после 10 эпох обучения (92.66%). По завершении 5 эпох предобученная сеть показала точность 93.69%, что на 1.03% превышает результат кастомной архитектуры.

Можно сделать вывод, что предобученная модель с более сложной архитектурой обладает лучшей способностью к быстрой адаптации и демонстрирует более высокую эффективность при одинаковых условиях обучения.

## 3. SOTA-результаты для выборки:

### 4. **State-of-the-art: ~96.7%**

Современные state-of-the-art методы достигают точности порядка 96.7% на тестовой выборке Fashion-MNIST. Разница в точности между нашими реализациями и SOTA-результатами обусловлена несколькими факторами:

**Архитектурные различия:** SOTA-методы используют специализированные архитектуры (PreActResNet18, EfficientNet) с оптимизированной структурой остаточных блоков

**Продолжительность обучения:** Наше обучение ограничено 5 эпохами, тогда как SOTA-результаты достигаются после длительного обучения с тщательным подбором гиперпараметров

**Техники аугментации:** Промышленные решения применяют более сложные методы аугментации данных

**Ансамблирование:** Лучшие результаты часто достигаются объединением нескольких моделей.

5. Визуализация работы СНС из пункта 1 (выбор и подачу на архитектуру произвольного изображения с выводом результата).

ResNet18:



CNN из лабораторной работы 1:



Вывод: Обе реализованные архитектуры показывают высокие результаты с разницей всего 1.03% в пользу предобученной ResNet18. Кастомная CNN демонстрирует отличную эффективность для относительно простой архитектуры, в то время как предобученная ResNet18 подтверждает преимущества трансферного обучения, достигая сопоставимой точности значительно быстрее.

Обе модели существенно уступают современным SOTA-методам (~3% разницы), что объясняется ограниченными вычислительными ресурсами и упрощенным подходом к обучению по сравнению с промышленными решениями. Для достижения SOTA-результатов потребовалось бы значительно большее время обучения, оптимизация гиперпараметров и применение продвинутых техник аугментации данных.