|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий

Кафедра Вычислительной Техники (ВТ)

**Отчёт**

по дисциплине

«Архитектура устройств и систем вычислительной техники»

|  |  |
| --- | --- |
|  | |
| Выполнили студенты группы ИВМО-02-24  Принял преподаватель | Кутепов А.О.  Рьянов А.Е.  Смирнов А.В.    Гуличева А.А. |
|  |  |

Работа выполнена «\_\_\_»\_\_\_\_\_\_\_\_20\_\_\_г

«Зачтено»

«\_\_\_»\_\_\_\_\_\_\_\_20\_\_\_г

Москва 2024

Оглавление

[Реализация и обучение НС для задачи классификации 2](#_Toc183082910)

[Реализация и обучение НС для задачи регрессии 5](#_Toc183082911)

[Реализация и обучение НС для распознавания изображений 11](#_Toc183082912)

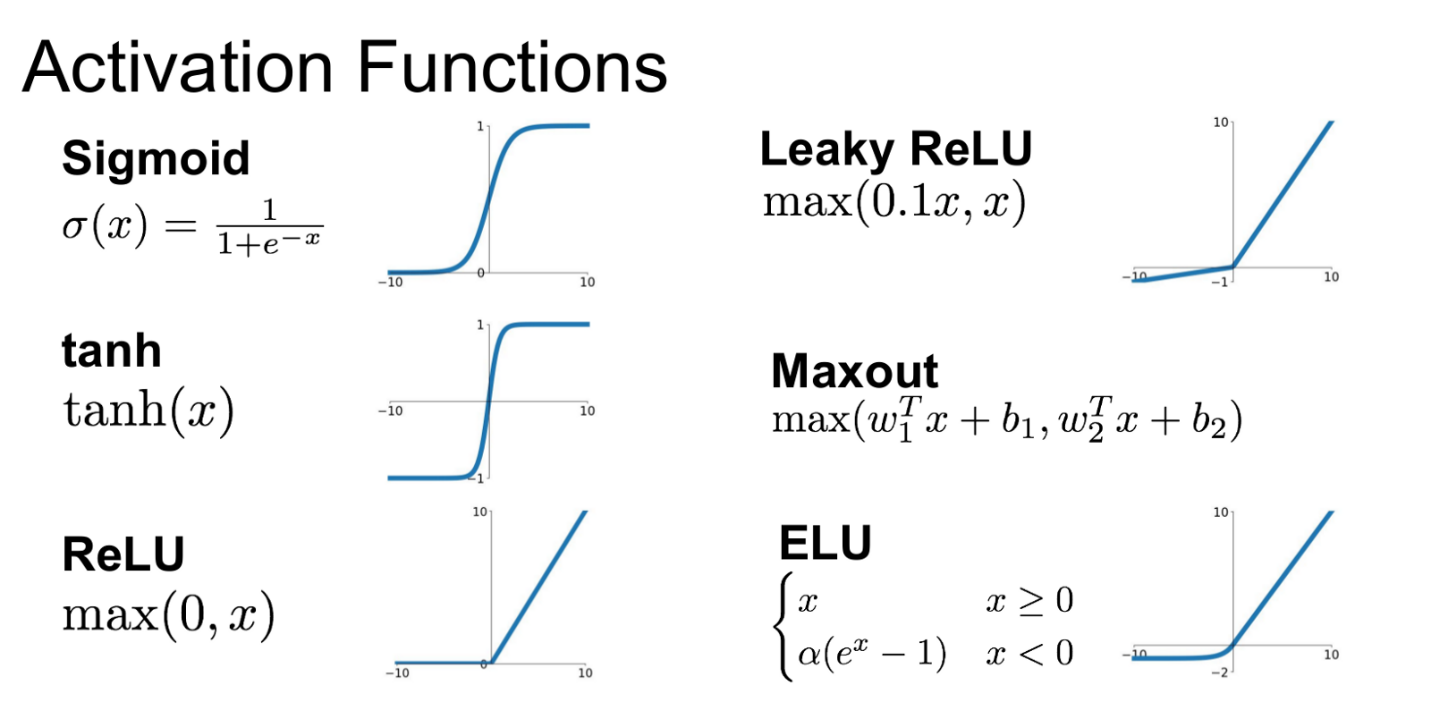
# Реализация и обучение НС для задачи классификации

Была разработана нейронная сеть для классификации рака груди на основе признаков из датасета <https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data?resource=download&select=data.csv>

Для обучения нейронов в Keras используются различные функции активации, такие как:

* softmax,
* elu,
* relu,
* tanh,
* sigmoid,
* softsign.

Эти функции активации применяются через слои или аргументы активации в функциях Dense() и Activation().



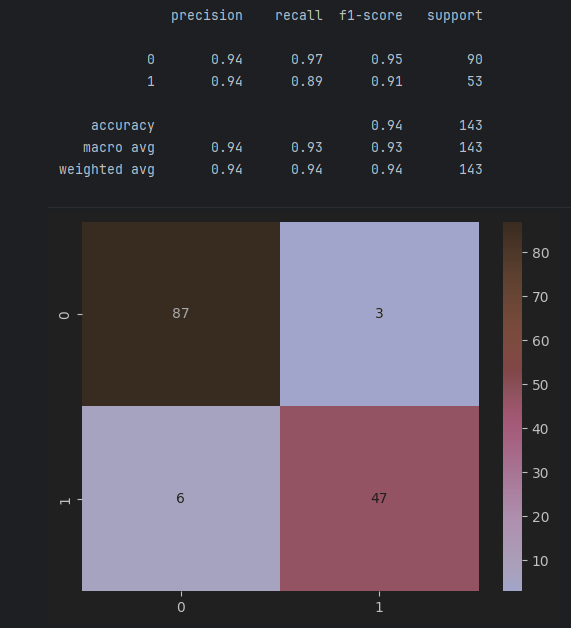
Для создания нейронных сетей в Keras используют следующие слои:

1. Dense — основной слой, который вычисляет вывод с использованием активации, ядра и смещения.
2. Dropout — слой, применяемый для борьбы с переобучением, случайным образом устанавливает некоторые единицы в 0.
3. Flatten — слой, преобразующий данные в меньшую размерность для подачи на вход следующего слоя.
4. Input — слой, используемый для создания модели на основе ввода и вывода модели.
5. Reshape — слой, изменяющий форму вывода на заданную размерность.
6. Permute — слой, меняющий порядок измерений в данных.
7. Lambda — слой, позволяющий создавать дополнительные признаки, отсутствующие в Keras.
8. Conv1D и Conv2D — слои для создания свёрточных сетей.
9. MaxPooling1D и MaxPooling2D — слои для уменьшения размера ввода и извлечения важной информации.
10. AveragePooling1D и AveragePooling2D — слои для извлечения среднего значения из данных.
11. SimpleRNN и LSTM — рекуррентные слои для вычисления последовательных данных.

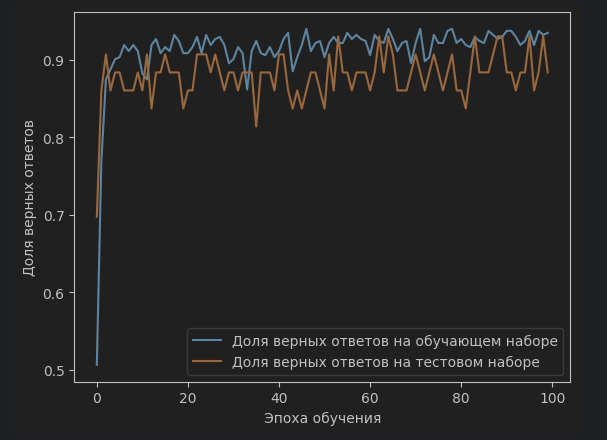
Гиперпараметры модели НС включают:

1. Количество скрытых слоев(layers) — определяет количество внутренних слоев в нейронной сети.
2. Количество нейронов в каждом скрытом слое(units) — указывает число узлов в каждом внутреннем слое.
3. Функции активации(activation) — выбирает функцию, которая преобразует выходные данные предыдущего слоя перед передачей их на следующий слой.
4. Скорость обучения(learning\_rate) — определяет, насколько быстро модель будет обучаться на обучающих данных.
5. Коэффициент отсева(dropout) — контролирует вероятность того, что нейроны будут случайно отключены во время обучения.
6. Количество эпох(epoch) — указывает, сколько раз модель будет проходить через обучающие данные.
7. Размер батча (batch size) — это общее число тренировочных объектов, представленных в одном батче. Батчи используются для эффективного обучения больших объёмов данных, так как позволяют разделить данные на части меньшего размера и загружать их по очереди.
8. Оптимизатор(optimizer) — оптимизирует функцию потерь для нахождения наилучших весов для прогнозирования. В Keras доступны различные типы оптимизаторов, такие как SGD, RMSProp, Adam, Adadelta, Adagrad, Adamax, Nadam и FTRL.
9. Функция ошибки(loss) в контексте машинного обучения — это функция, которая оценивает качество работы модели и пытается минимизировать ошибку во время обучения.

Точность обученной модели:

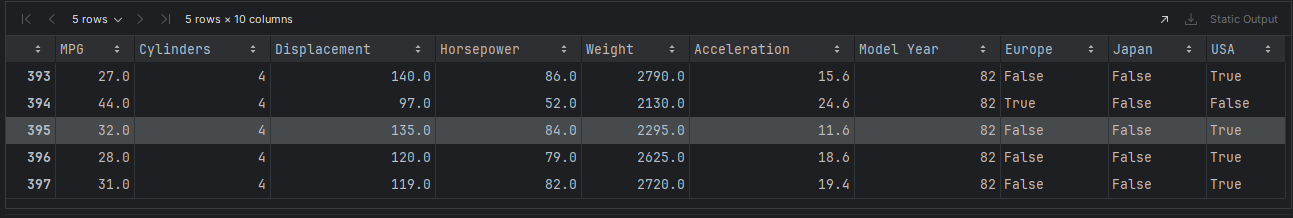


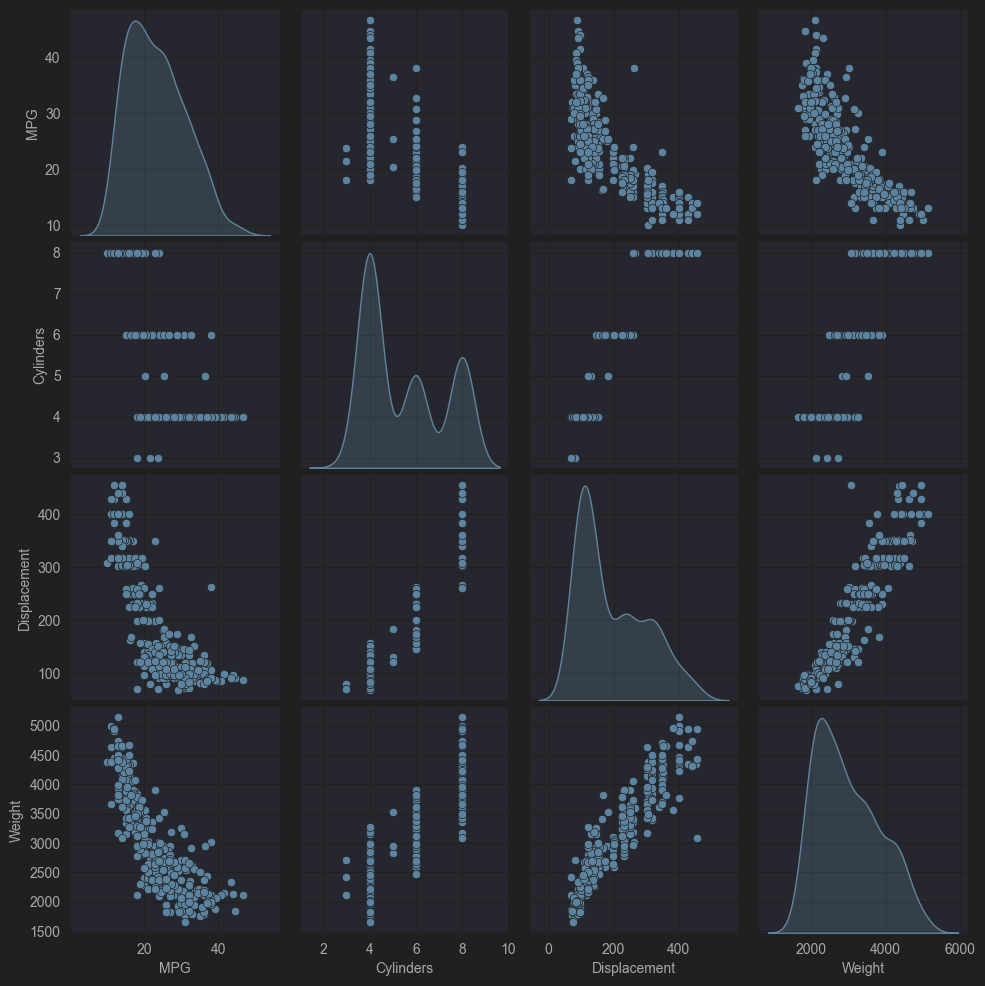
Время выполнения: 5.43s



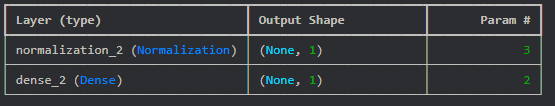
# Реализация и обучение НС для задачи регрессии

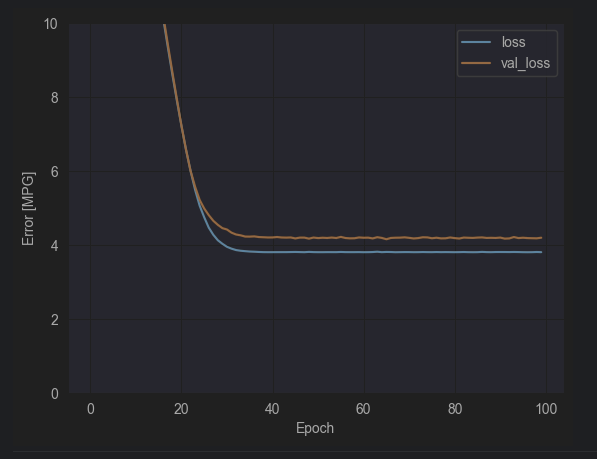
Построены различные модели регрессии для прогнозирование эффективности использования топлива на основе признаков из датасета <https://archive.ics.uci.edu/dataset/9/auto+mpg>

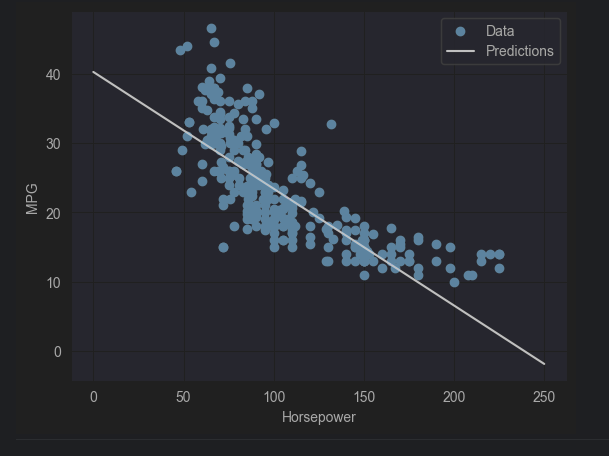




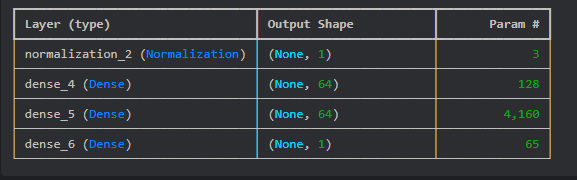
**Линейная регрессия с одной переменной. Предсказание `MPG` по `Horsepower`**

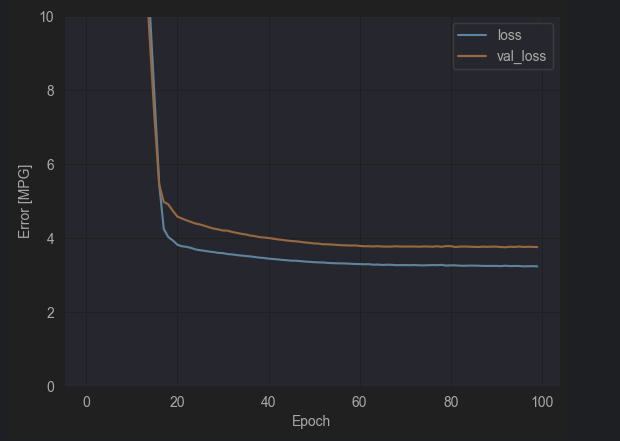


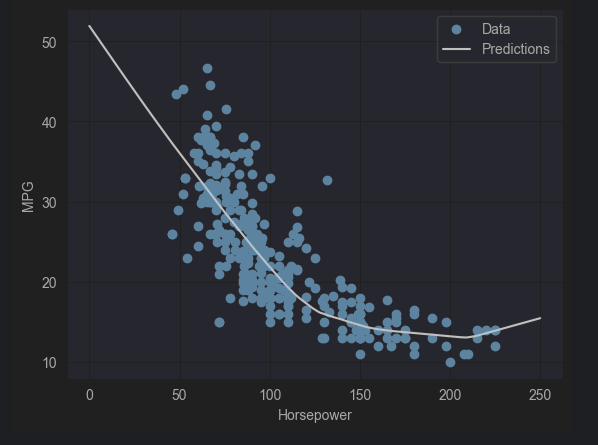




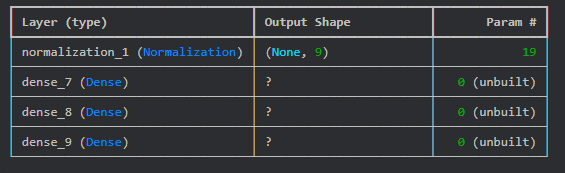
**Регрессия с глубокой нейронной сетью (DNN)**

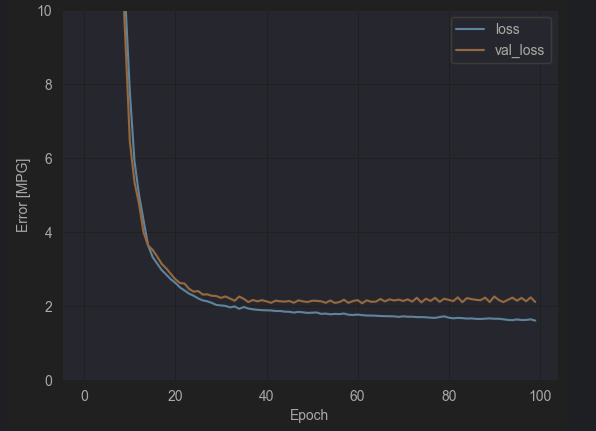


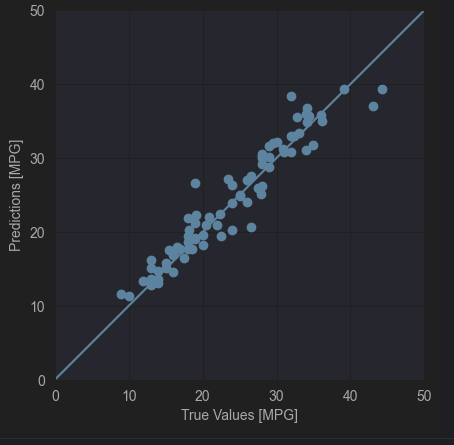




**Регрессия с использованием DNN и нескольких входных данных**



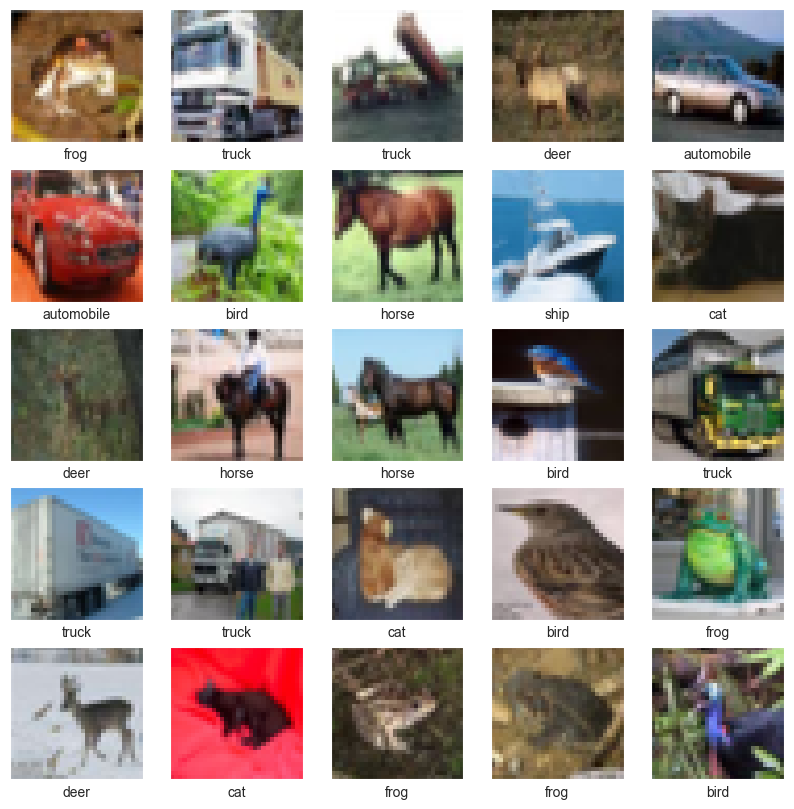


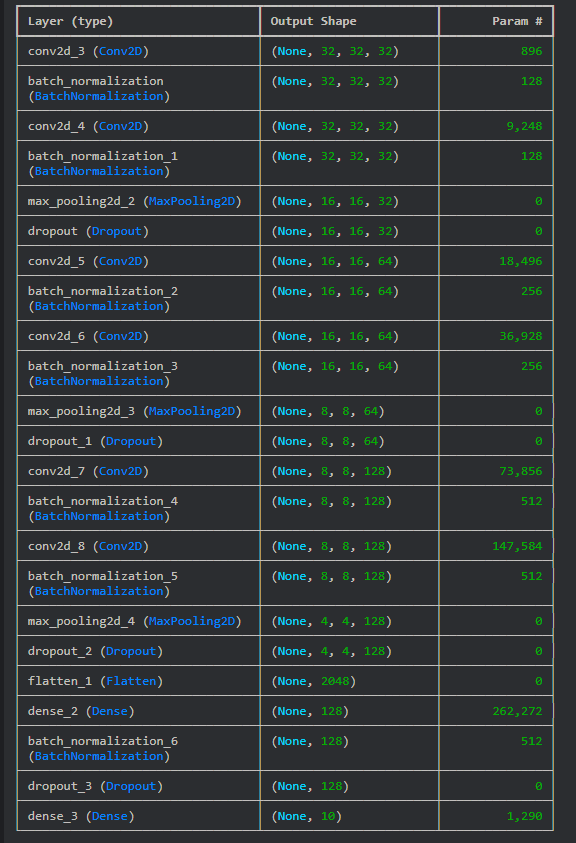


# Реализация и обучение НС для распознавания изображений

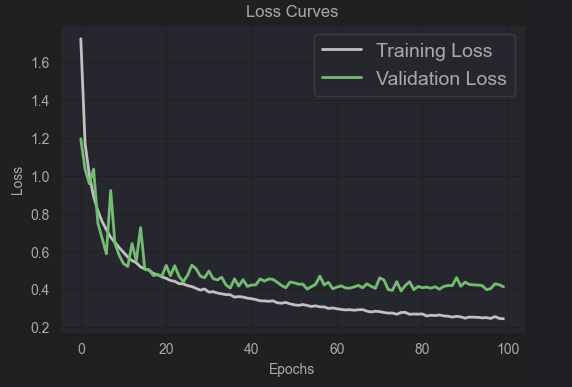
Реализована нейронная сеть для классификации изображений на основе датасетов CIFAR-10 и CIFAR-100

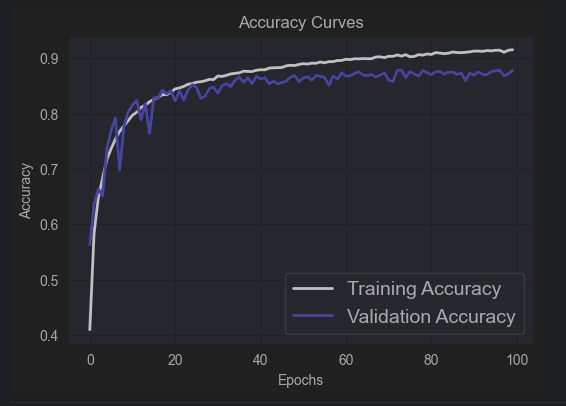
**CIFAR-10**

****

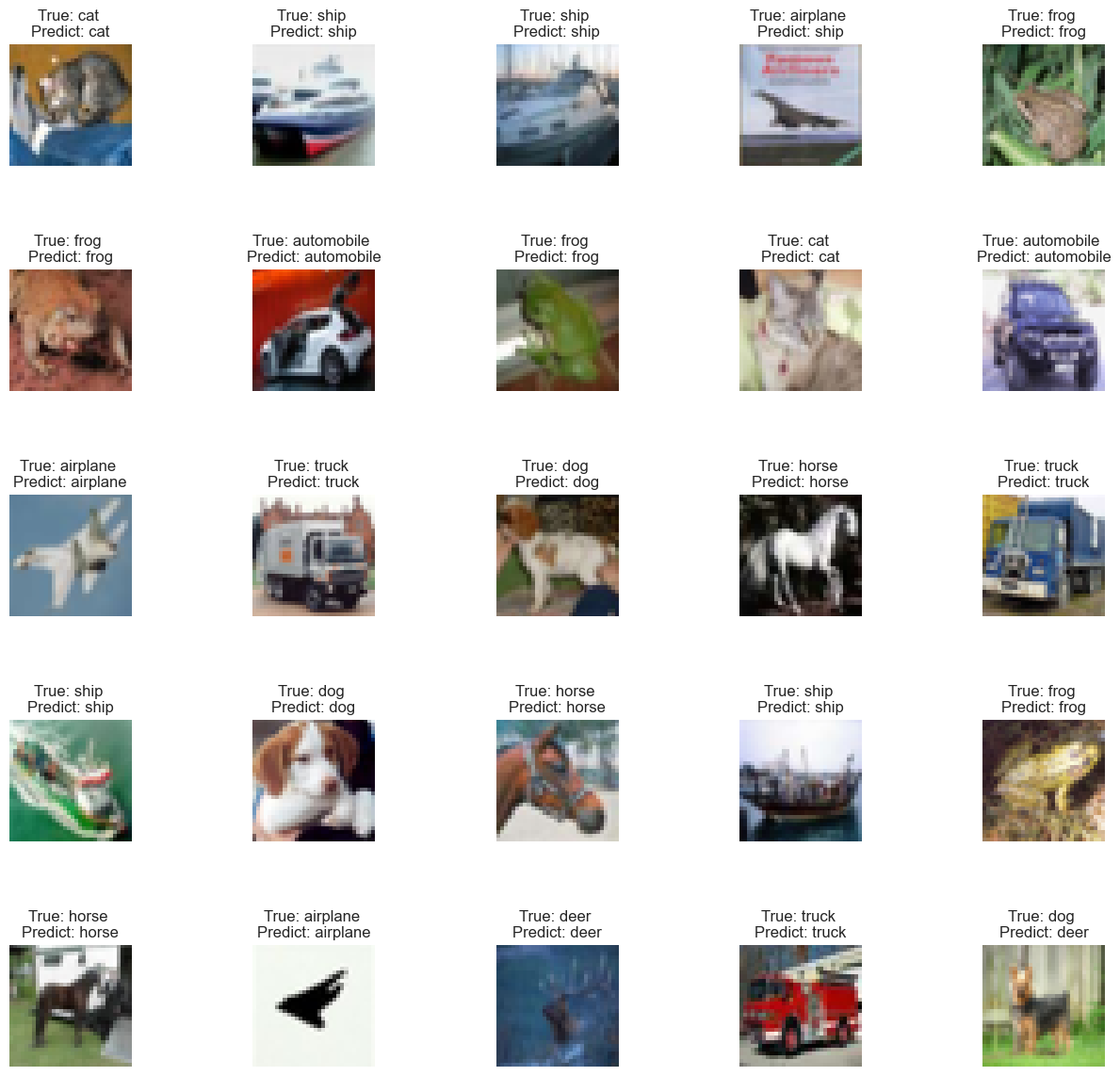
****

Время обучения: 16h 43min 8s

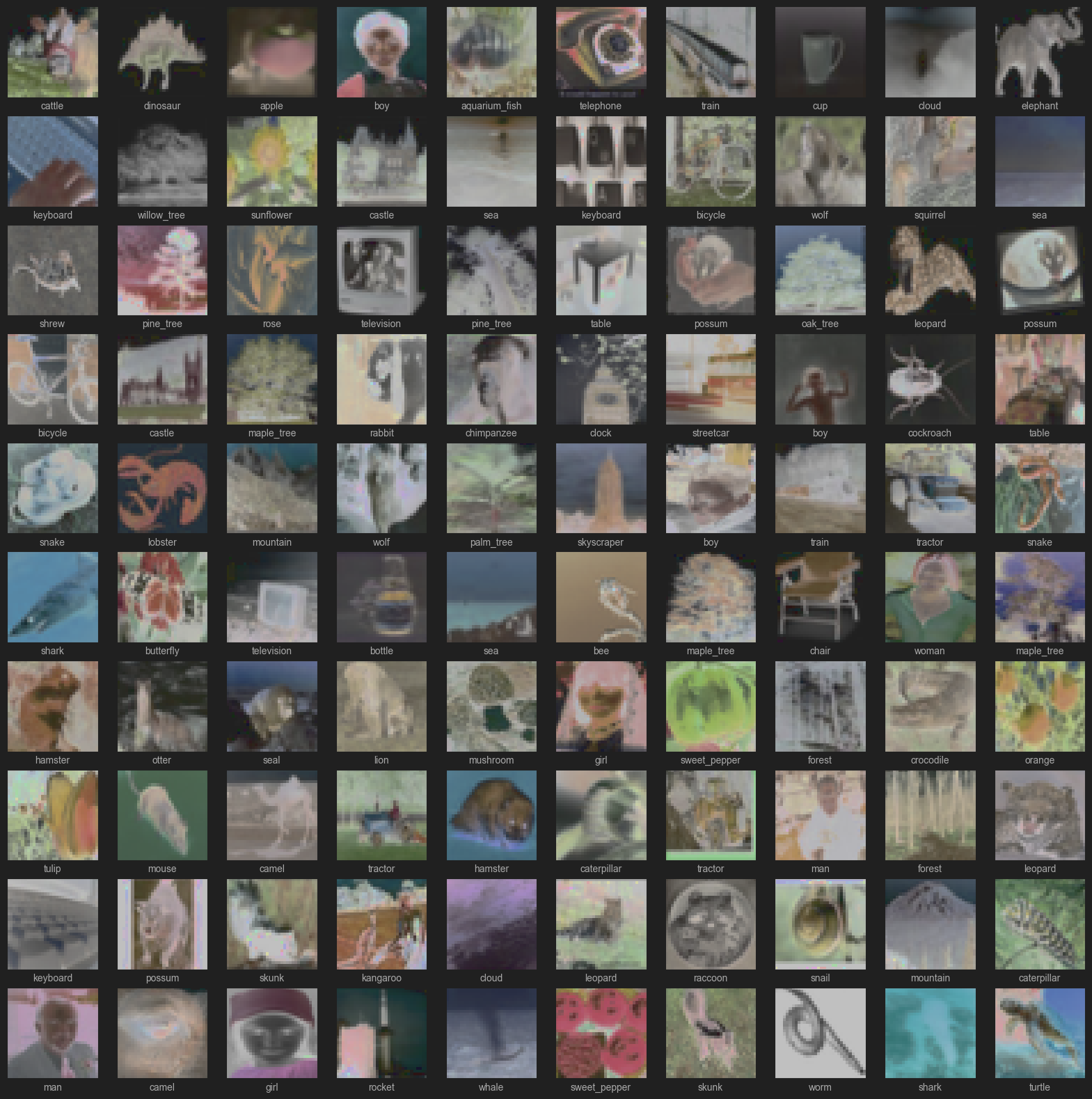




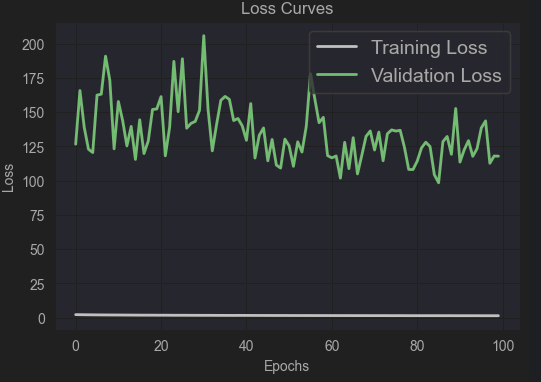
Предсказания:

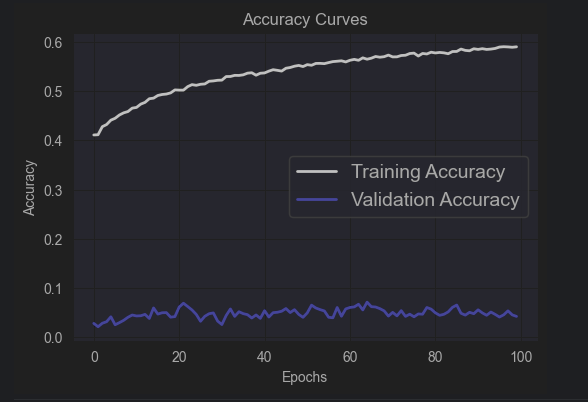


**Cifar-100**

****

****

****

****

Предсказания:

