

МО 2

Приложил титанические усилия для выполнение проекта: Беляев Андрей Евгеньевич

Постановка задачи

Цель проекта

Распознать изображение с картинки и определить к какому из 10 классов она относится.

Вход

Изображение с одним из 10-ти возможных объекта

Выход

Имя класса, к которому объект относится

Общая информация

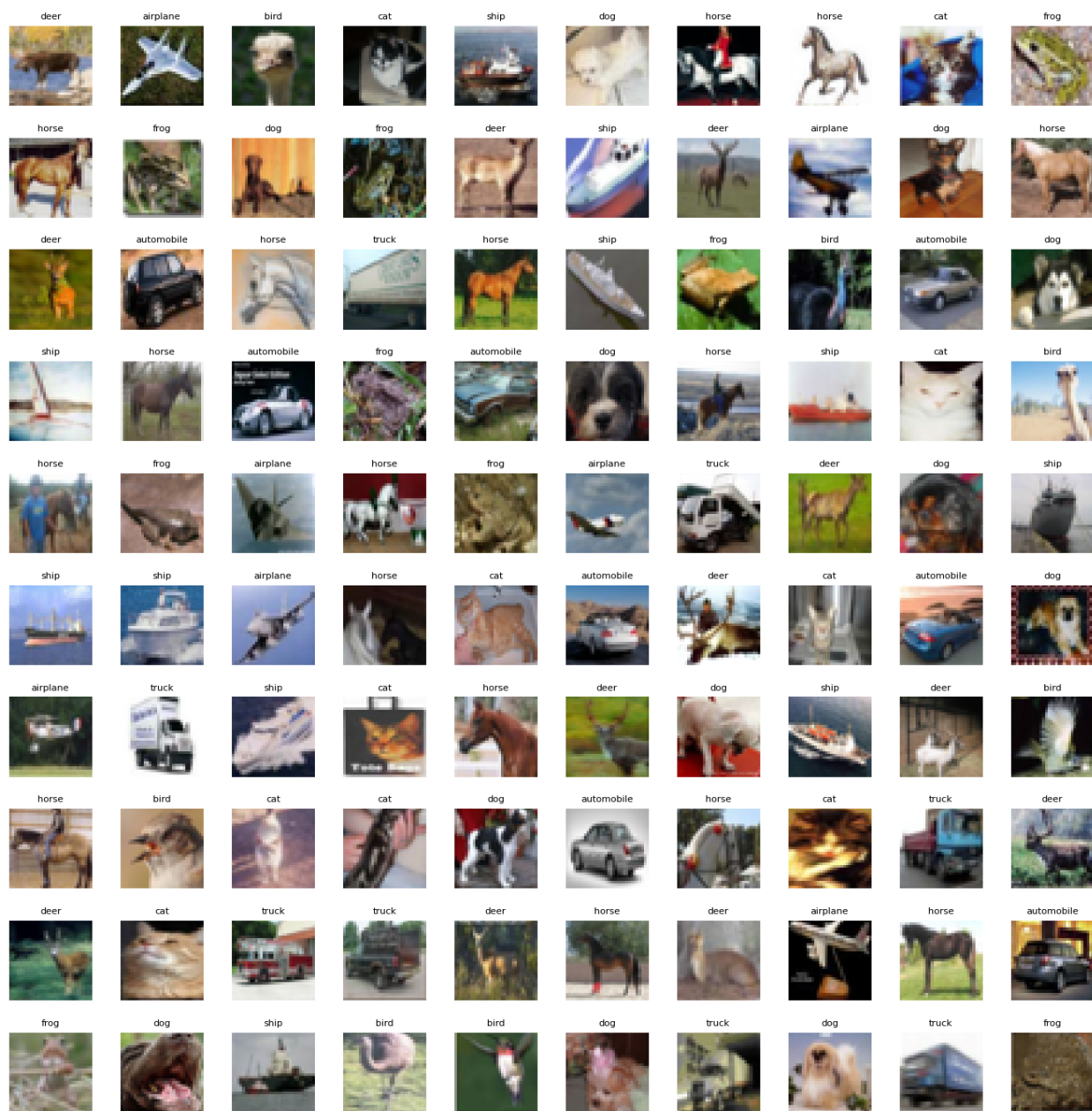
Распознавание изображения относится к задаче ввода изображения в нейронную сеть и присвоения какой-либо метки для этого изображения. Метка, которую выводит сеть, будет соответствовать заранее определенному классу. Может быть присвоено как сразу несколько классов, так и только один. Если существует всего только один класс, обычно применяется термин «распознавание», тогда как задача распознавания нескольких классов часто называется «классификацией».

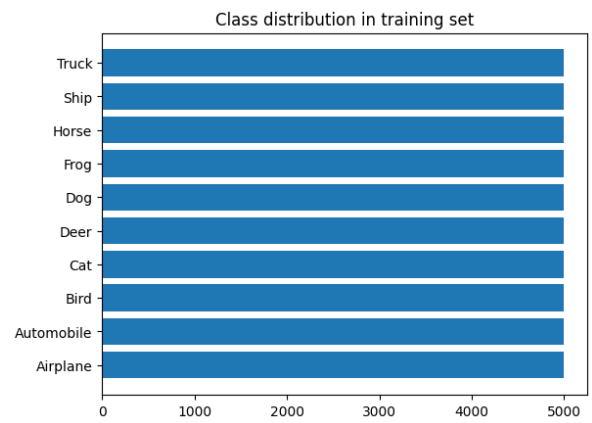
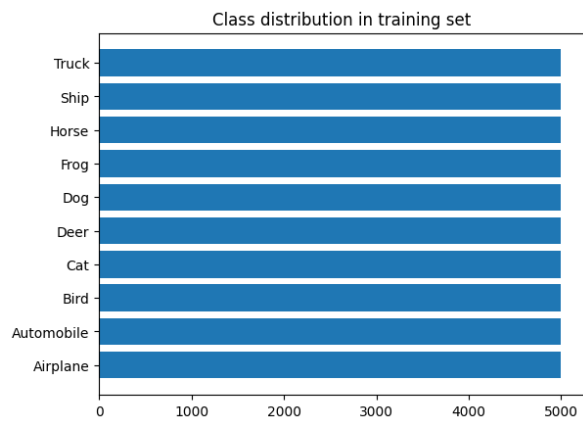
Чтобы выполнить распознавание или классификацию изображений, нейронная сеть должна выполнить извлечение признаков. Признаки - это элементы данных, которые представляют максимальный интерес и которые будут передаваться по нейросети. В конкретном случае распознавания

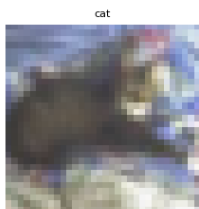
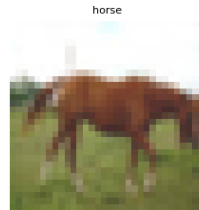
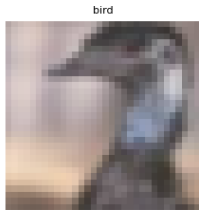
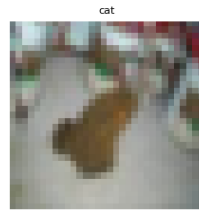
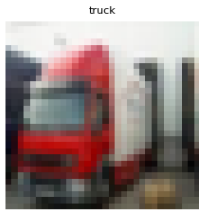
изображений такими признаками являются группы пикселей, такие как линии и точки, которые сеть будет анализировать на наличие паттерна.

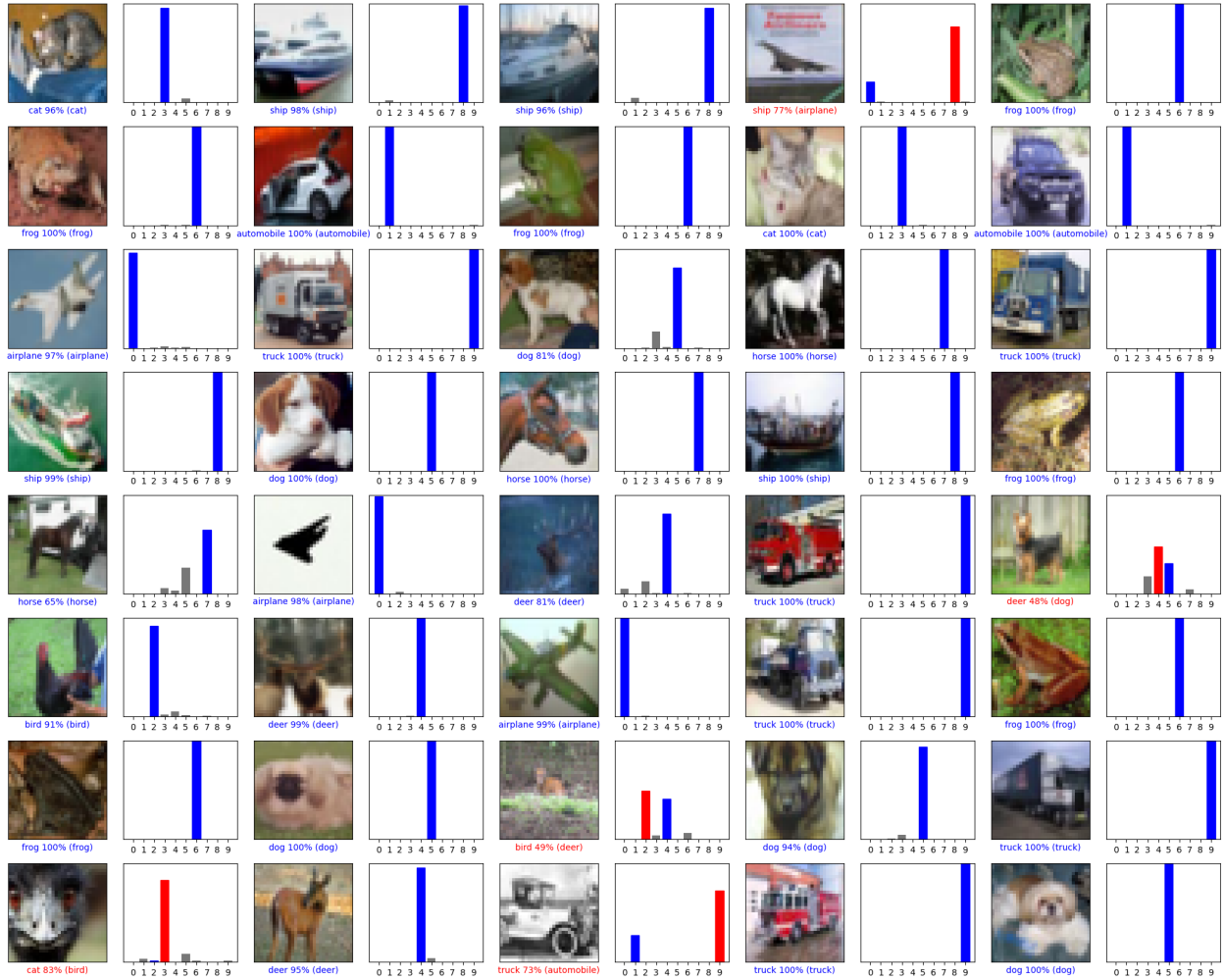
Информация о датасете

Набор данных CIFAR-10 содержит 60 000 цветных изображений размером 32 x 32 пикселя в 3 каналах, разделенных на 10 классов. Каждый класс содержит 6000 изображений. Обучающий набор содержит 50 000 изображений, а тестовый набор — 10 000 изображений.









Модель 😊

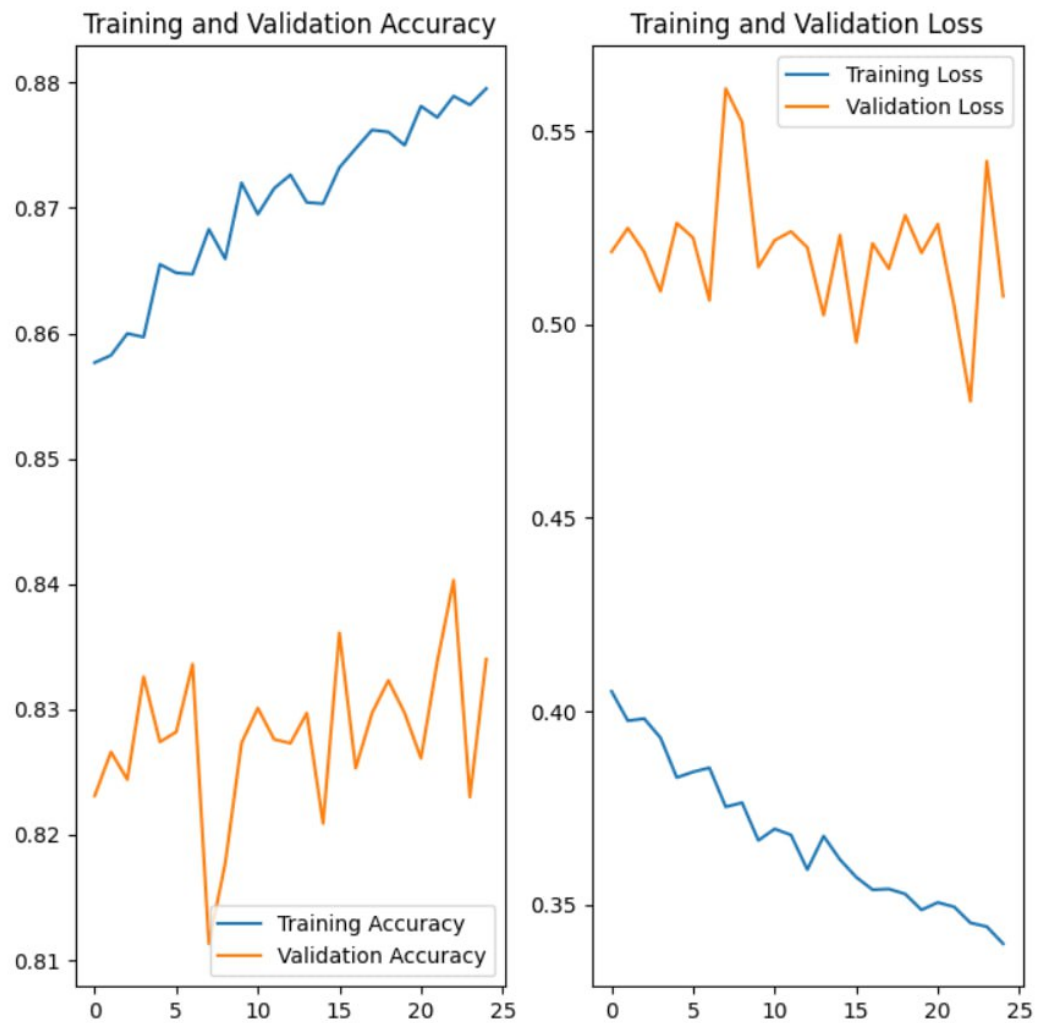
Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	896
batch_normalization (Batch Normalization)	(None, 32, 32, 32)	128
conv2d_1 (Conv2D)	(None, 32, 32, 32)	9248
batch_normalization_1 (Batch Normalization)	(None, 32, 32, 32)	128
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
dropout (Dropout)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	18496
batch_normalization_2 (Batch Normalization)	(None, 16, 16, 64)	256
conv2d_3 (Conv2D)	(None, 16, 16, 64)	36928
batch_normalization_3 (Batch Normalization)	(None, 16, 16, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 64)	0
dropout_1 (Dropout)	(None, 8, 8, 64)	0
conv2d_4 (Conv2D)	(None, 8, 8, 128)	73856
batch_normalization_4 (Batch Normalization)	(None, 8, 8, 128)	512
conv2d_5 (Conv2D)	(None, 8, 8, 128)	147584
batch_normalization_5 (Batch Normalization)	(None, 8, 8, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout_2 (Dropout)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 128)	262272
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1290

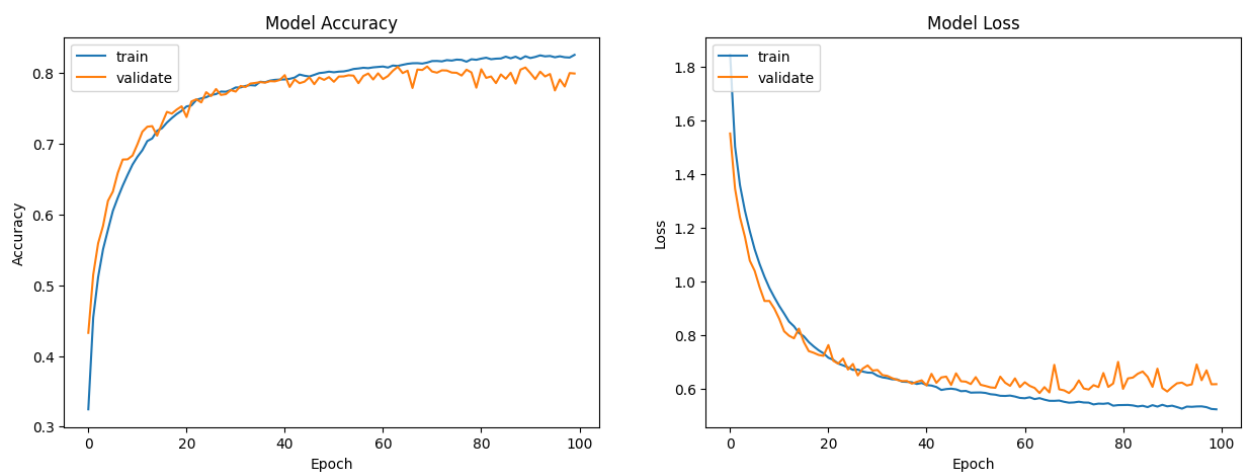
1. Создается модель типа Sequential, который представляет собой линейный стек слоев.
2. Первый блок
 - Добавляется слой Conv2D с 32 фильтрами размером (3, 3) и 'same' .
 - Применяется функция активации ReLU.
 - Добавляется еще один слой Conv2D с 32 фильтрами размером (3, 3).
 - Применяется функция активации ReLU.
 - Добавляется слой MaxPooling2D с пулом размером (2, 2) для уменьшения размерности.
 - Добавляется слой Dropout с коэффициентом 0.25 для предотвращения переобучения.
3. Второй блок
 - Аналогично первому блоку, но с увеличенным количеством фильтров (64)
4. FLATTEN ⇒ DENSE ⇒ RELU ⇒ DROPOUT
 - Выполняется операция Flatten для преобразования выхода сверточных слоев в одномерный вектор.
 - Применяется функция активации ReLU.
 - Добавляется полносвязный слой Dense с 512 нейронами.
 - Добавляется слой Dropout с коэффициентом 0.5.
5. Softmax классификатор
 - Добавляется полносвязный слой Dense с количеством нейронов, равным числу классов (num_classes).
 - Применяется функция активации Softmax для получения вероятностного распределения по классам.

Исследования

Первый вид модели



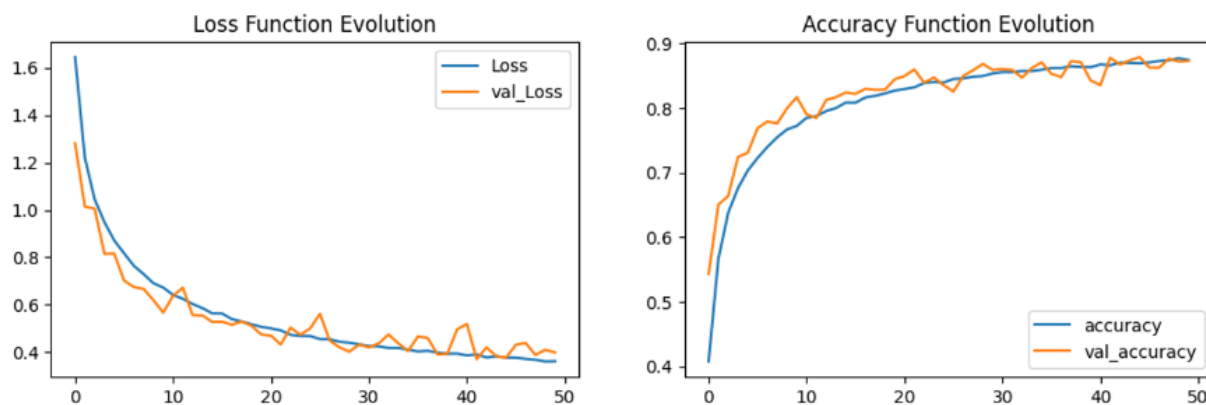
Строим более подходящую модель



Исходя из графика точности и графика потерь делаем вывод, что модель обучается успешно.

После 60 эпох точность нашей модели особо не увеличивается. Но наша модель не переобучается.

После изменения модели



1) График Loss Function Evolution отражает изменение функции потерь модели во время обучения.

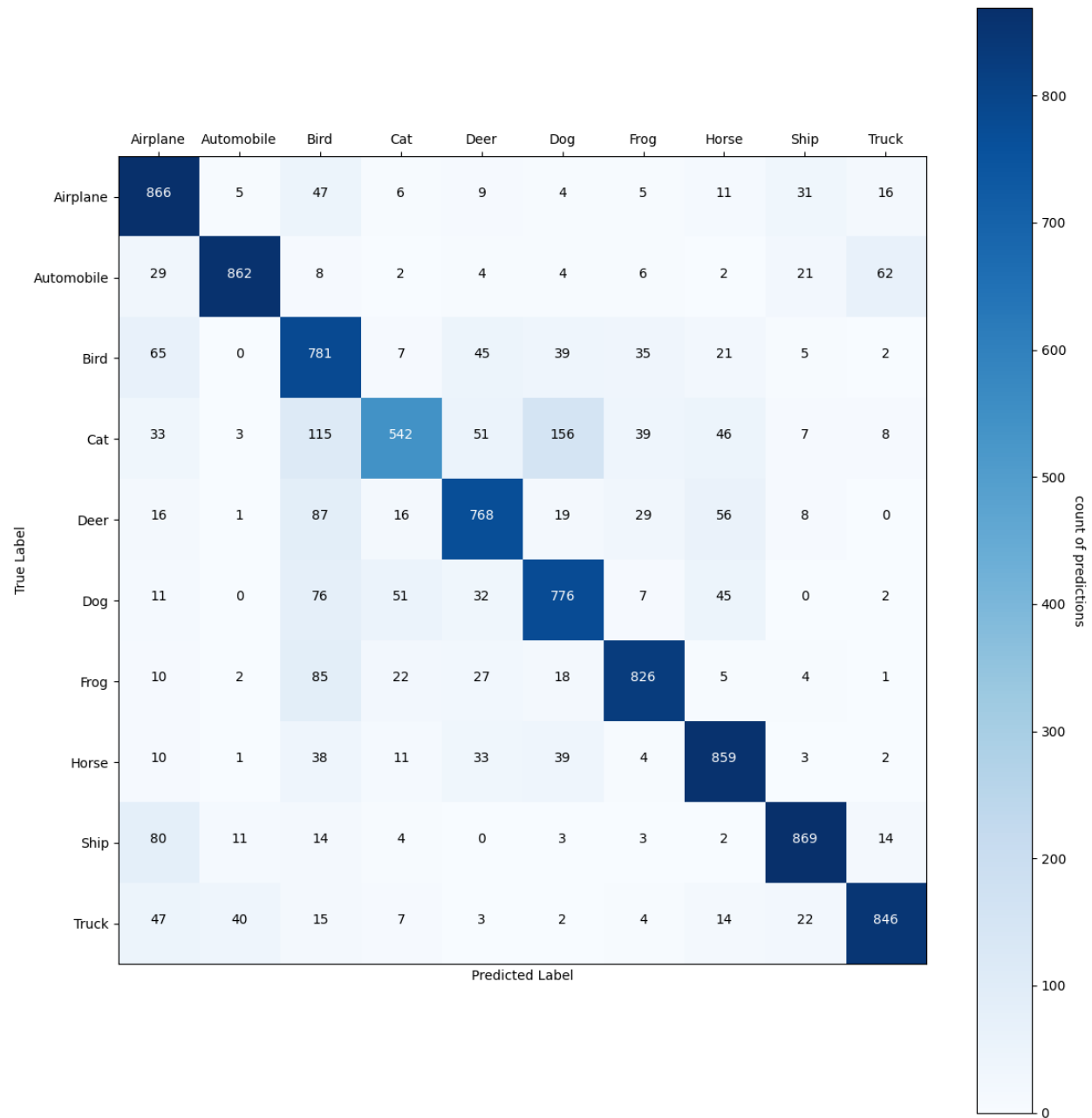
2) График Accuracy Function Evolution показывает изменение точности модели во время обучения.

3) График Precision Function Evolution отображает изменение точности модели в предсказании положительного класса. Точность (precision) вычисляется как отношение истинно положительных предсказаний к общему числу предсказаний положительного класса.

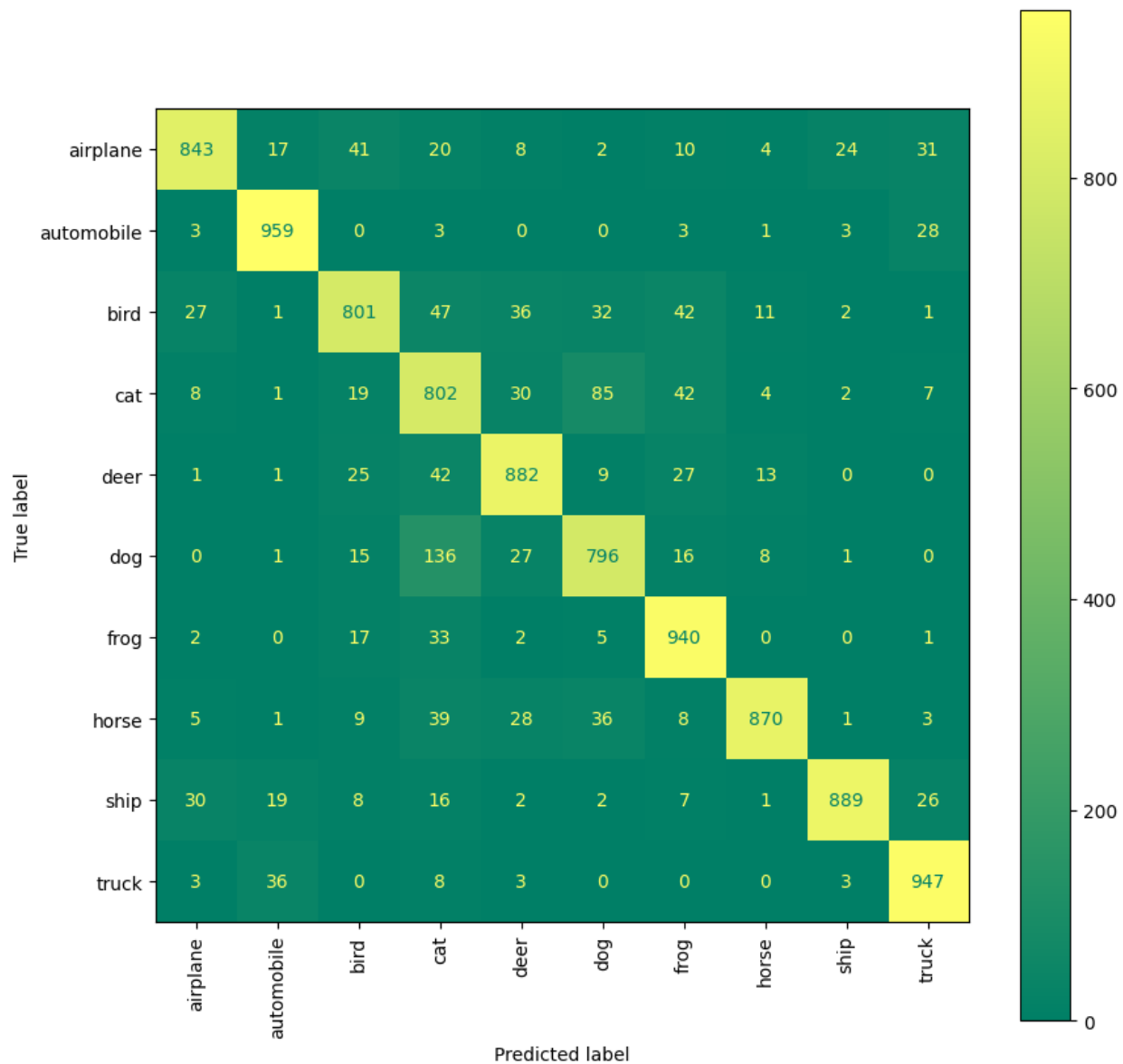
4) График Recall Function Evolution демонстрирует изменение полноты модели в предсказании положительного класса. Полнота (recall) вычисляется как отношение истинно положительных предсказаний к общему числу истинных значений положительного класса.

Давайте посмотрим в чем заключаются ошибки. Грубо говоря имеется две гипотезы, что точка роста находится в датасете/в построении модели.

Составим матрицу соответствия!



После изменения матрица выглядит так



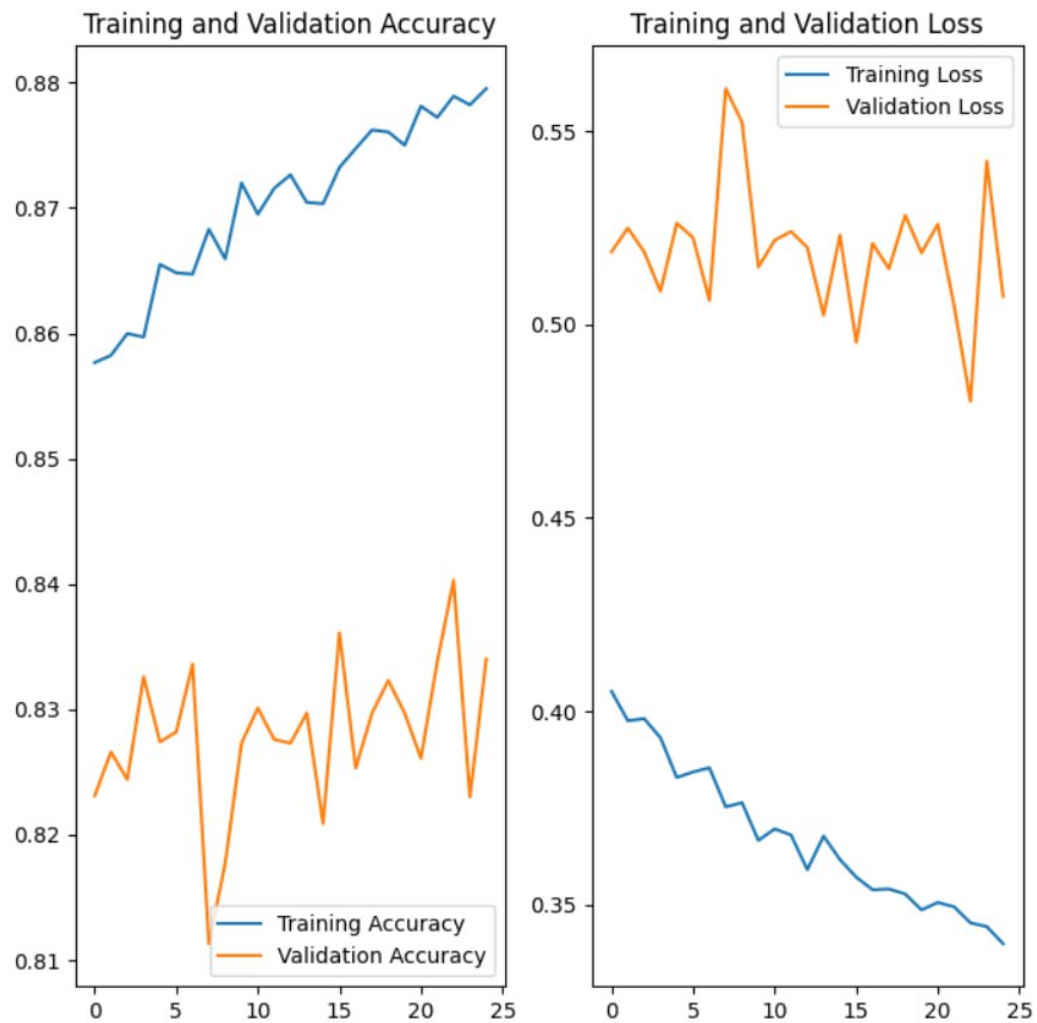
По вертикали обозначены реальный классы, к которым относятся фотографии. По горизонтали предсказание модели, при поступлении на вход такой же фотографии.

	precision	recall	f1-score	support
0	0.74	0.87	0.80	1000
1	0.93	0.86	0.90	1000
2	0.62	0.78	0.69	1000
3	0.81	0.54	0.65	1000
4	0.79	0.77	0.78	1000
5	0.73	0.78	0.75	1000
6	0.86	0.83	0.84	1000
7	0.81	0.86	0.83	1000
8	0.90	0.87	0.88	1000
9	0.89	0.85	0.87	1000
accuracy			0.80	10000
macro avg	0.81	0.80	0.80	10000
weighted avg	0.81	0.80	0.80	10000

Здесь мы видим 4 метрики для каждого класса:

- Точность (Precision): доля верно предсказанных положительных наблюдений от общего числа предсказанных положительных наблюдений.
- Полнота (Recall): доля верно предсказанных положительных наблюдений от общего числа истинных положительных наблюдений.
- F1-мера (F1-score): гармоническое среднее точности и полноты.
- Поддержка (Support): количество появлений каждого класса в истинных метках.

Из этого исследования мы видим, что кошки являются самыми проблемными для определения.



Также на этих фотографиях мы можем убедиться в правильности гипотез по входным данным.

Ссылки

<https://github.com/Jedykqoo/ml>