МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Искусственные нейронные сети»

Тема: «Распознавание рукописных символов»

Студентка гр. 8382 _____ Звегинцева Е.Н.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы.

Реализовать классификацию черно-белых изображений рукописных цифр (28x28) по 10 категориям (от 0 до 9). Набор данных содержит 60 000 изображений для обучения и 10 000 изображений для тестирования.

Задачи.

- Ознакомиться с представлением графических данных
- Ознакомиться с простейшим способом передачи графических данных нейронной сети
- Создать модель
- Настроить параметры обучения
- Написать функцию, позволяющая загружать изображение пользователи и классифицировать его

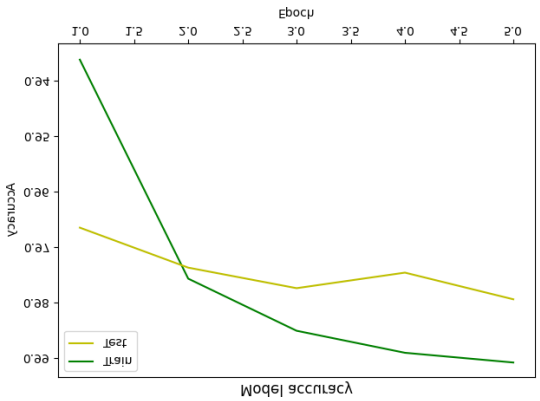
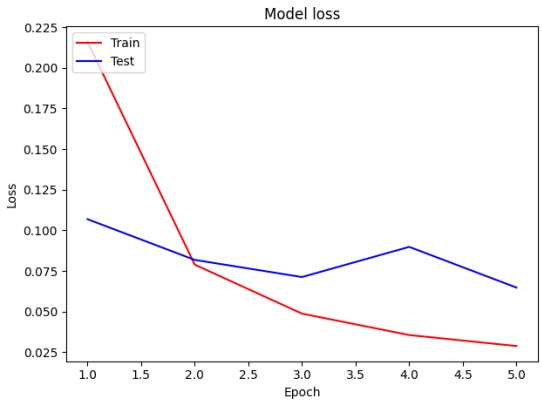
Требования.

- Найти архитектуру сети, при которой точность классификации будет не менее 95%
- Исследовать влияние различных оптимизаторов, а также их параметров, на процесс обучения
- Написать функцию, которая позволит загружать пользовательское изображение не из датасета

Ход работы.

В ходе работы была создана и обучена модель искусственной нейронной сети в соответствии с условиями (код представлен в приложении).

Результатом поиска наилучшей архитектуры, при которой точность классификации сети более 95%, оказалась модель с 2мя скрытыми слоями на 512 и 216 нейронов. На самом деле добавление слоев не сильно влияет на результат, но при таком строении потери наименьшие. Эти результаты достигнуты при исходном количестве эпох = 5, однако в процессе тестирования я выяснила, что точнее результаты при количестве эпох = 20.

График точности	График потерь
	

Результаты обучения модели при использовании различных оптимизаторов, а также их параметров представлены в таблицах ниже. Чтобы определить какой оптимизатор и с какими параметрами подходит нам больше всего, я рассмотрела каждый оптимизатор с различными параметрами. Далее из них выбирала наилучший и фиксировала, что менялось.

Название оптимизатора	Его параметры	Точность
Adadelta	'learning_rate': 0.001, 'decay': 0.0, 'rho': 0.95, 'epsilon': 1e-07}:	0.8683
	'learning_rate': 0.001, 'decay': 0.0, 'rho': 0.5, 'epsilon': 1e-07}:	0.8038
	learning_rate': 0.01, 'decay': 0.0, 'rho': 0.95, 'epsilon': 1e-07}:	0.9411

С увеличением 'learning_rate' увеличивается точность, поэтому наилучший вариант это:

{'name': 'Adadelta', 'learning_rate': 0.01, 'decay': 0.0, 'rho': 0.95, 'epsilon': 1e-07}: 0.9411

Название оптимизатора	Его параметры	Точность
Adagrad	'learning_rate': 0.001, 'decay': 0.0, 'initial_accumulator_value': 0.1, 'epsilon': 1e-07}	0.9285
	'learning_rate': 0.01, 'decay': 0.0, 'initial_accumulator_value': 0.1, 'epsilon': 1e-07}	0.9736

С увеличением 'learning_rate' увеличивается точность, поэтому наилучший вариант это:

{'name': 'Adagrad', 'learning_rate': 0.01, 'decay': 0.0, 'initial_accumulator_value': 0.1, 'epsilon': 1e-07}: 0.9736

Название оптимизатора	Его параметры	Точность
Adam	'learning_rate': 0.001, 'decay': 0.0, 'beta_1': 0.9, 'beta_2': 0.999, 'epsilon': 1e-07, 'amsgrad': False}	0.9838
	'learning_rate': 0.01, 'decay': 0.0, 'beta_1': 0.9, 'beta_2': 0.999, 'epsilon': 1e-07, 'amsgrad': False}:	0.9749
	{'name': 'Adamax', 'learning_rate': 0.001, 'decay': 0.0, 'beta_1': 0.9, 'beta_2': 0.999, 'epsilon': 1e-07}:	0.9825

С увеличением 'learning_rate' увеличивается потери, поэтому наилучший вариант это:

{'name': 'Adam', 'learning_rate': 0.001, 'decay': 0.0, 'beta_1': 0.9, 'beta_2': 0.999, 'epsilon': 1e-07, 'amsgrad': False}: 0.9838

Название оптимизатора	Его параметры	Точность
RMSprop	'learning_rate': 0.001, 'decay': 0.0, 'rho': 0.9, 'momentum': 0.0, 'epsilon': 1e-07, 'centered': False}	0.9828
	'learning_rate': 0.01, 'decay': 0.0, 'rho': 0.9, 'momentum': 0.0, 'epsilon': 1e-07, 'centered': False}	0.9644

	'learning_rate': 0.001, 'decay': 0.0, 'rho': 0.5, 'momentum': 0.0, 'epsilon': 1e-07, 'centered': False}	0.9823
	'learning_rate': 0.001, 'decay': 0.0, 'rho': 0.9, 'momentum': 0.9, 'epsilon': 1e-07, 'centered': False}:	0.9768

С увеличением параметров'learning_rate' и 'momentum' увеличивается потери, при изменении параметра 'rho' ничего не меняется, поэтому наилучший вариант это:

{'name': 'RMSprop', 'learning_rate': 0.001, 'decay': 0.0, 'rho': 0.9, 'momentum': 0.0, 'epsilon': 1e-07, 'centered': False}: 0.9828

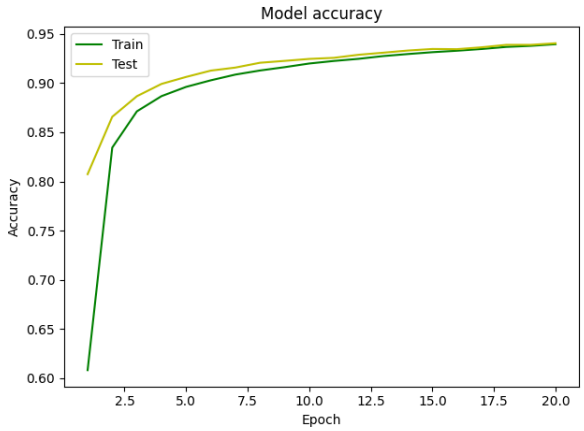
Название оптимизатора	Его параметры	Точность
SGD	'learning_rate': 0.01, 'decay': 0.0, 'momentum': 0.0, 'nesterov': False}	0.9568
	'learning_rate': 0.001, 'decay': 0.0, 'momentum': 0.0, 'nesterov': False}	0.9014
	'learning_rate': 0.01, 'decay': 0.0, 'momentum': 0.9, 'nesterov': False}:	0.9799

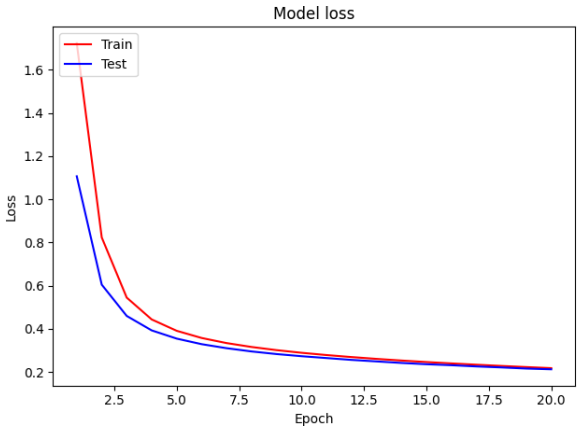
С увеличением 'learning_rate' и 'momentum' увеличивается точность, поэтому наилучший вариант это:

{'name': 'SGD', 'learning_rate': 0.01, 'decay': 0.0, 'momentum': 0.9, 'nesterov': False}: 0.9799

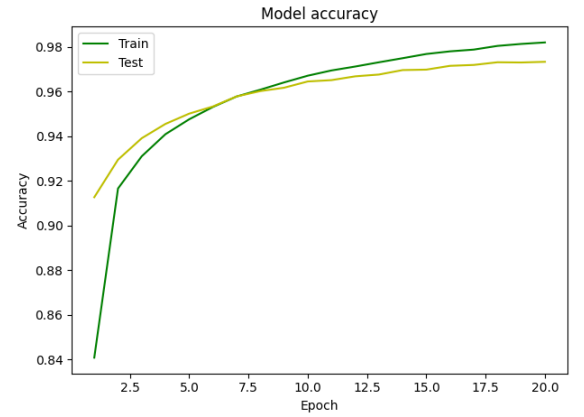
Теперь из наилучших вариаций оптимизаторов выберем самый подходящий:

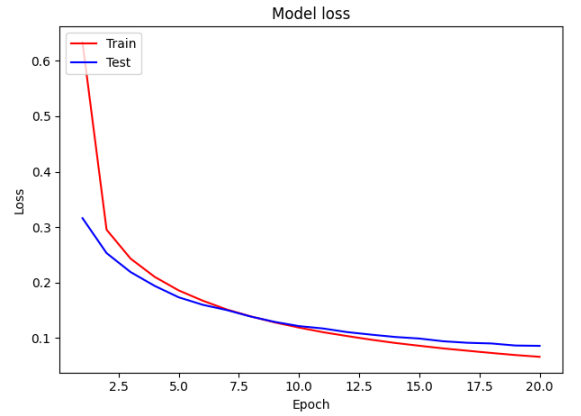
{'name': 'Adadelta', 'learning_rate': 0.01, 'decay': 0.0, 'rho': 0.95, 'epsilon': 1e-07}: 0.9404



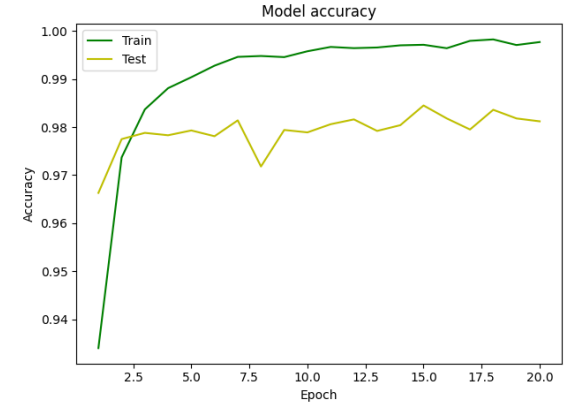


{'name': 'Adagrad', 'learning_rate': 0.01, 'decay': 0.0, 'initial_accumulator_value': 0.1, 'epsilon': 1e-07}: 0.9733

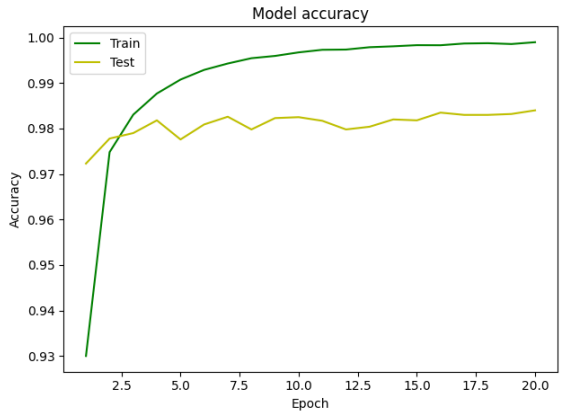


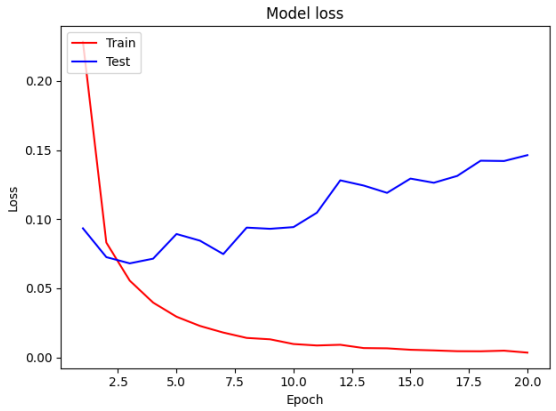


{'name': 'Adam', 'learning_rate': 0.001, 'decay': 0.0, 'beta_1': 0.9, 'beta_2': 0.999, 'epsilon': 1e-07, 'amsgrad': False}: 0.9812

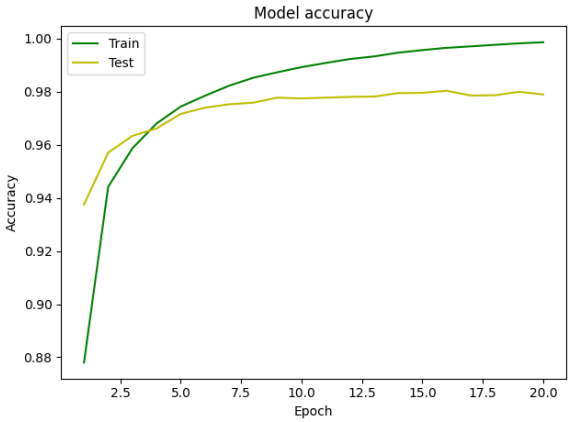


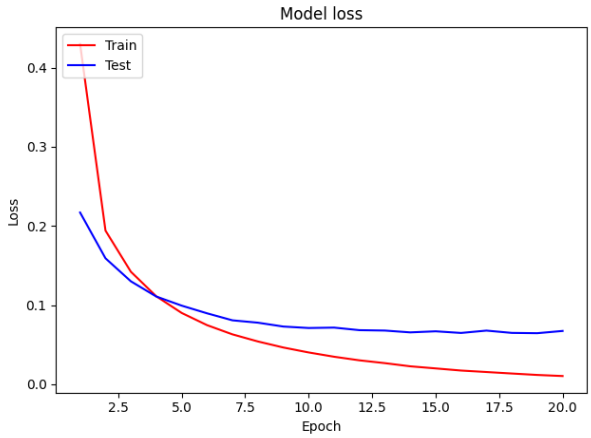
{'name': 'RMSprop', 'learning_rate': 0.001, 'decay': 0.0, 'rho': 0.9, 'momentum': 0.0, 'epsilon': 1e-07, 'centered': False}: 0.984





{'name': 'SGD', 'learning_rate': 0.01, 'decay': 0.0, 'momentum': 0.9, 'nesterov': False}: 0.9789





Сравнивая полученные цифры и смотря на графики, можно увидеть, что хуже всех себя оптимизатор `'Adadelta'`, так как у нее самая маленькая точность. Лучше всех справился оптимизатор `'RMSprop'`.

Запуск программы с изображением пользователя.

Рисунок 1 – файл q.png

Вывод программы:

test_acc: 0.9818000197410583

8

Выводы.

В ходе выполнения данной работы была изучена задача распознавания рукописных цифр и исследовано влияние различных оптимизаторов на обучение моделей. Также была произведена работа по работе и обработке изображений.