

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет	Компьютерных сетей и систем
Кафедра	Информатики

МАШИННОЕ ОБУЧЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА №2
«Реализация глубокой нейронной сети»

БГУИР 1-40 81 04

Магистрант:
гр. 858641
Кукареко А.В.

Проверил:
Стержанов М. В.

Минск, 2020

ХОД РАБОТЫ

Данные.

В работе предлагается использовать набор данных notMNIST, который состоит из изображений размерностью 28×28 первых 10 букв латинского алфавита (A ... J, соответственно). Обучающая выборка содержит порядка 500 тыс. изображений, а тестовая – около 19 тыс.

Задание.

1. Реализуйте полносвязную нейронную сеть с помощью библиотеки Tensor Flow. В качестве алгоритма оптимизации можно использовать, например, стохастический градиент (Stochastic Gradient Descent, SGD). Определите количество скрытых слоев от 1 до 5, количество нейронов в каждом из слоев до нескольких сотен, а также их функции активации (кусочно-линейная, сигмоидная, гиперболический тангенс и т.д.).
2. Как улучшилась точность классификатора по сравнению с логистической регрессией?
3. Используйте регуляризацию и метод сброса нейронов (dropout) для борьбы с переобучением. Как улучшилось качество классификации?.
4. Воспользуйтесь динамически изменяемой скоростью обучения (learning rate). Наилучшая точность, достигнутая с помощью данной модели составляет 97.1%. Какую точность демонстрирует Ваша реализованная модель?

Результат выполнения:

1. Реализуйте полносвязную нейронную сеть с помощью библиотеки Tensor Flow. В качестве алгоритма оптимизации можно использовать, например, стохастический градиент (Stochastic Gradient Descent, SGD). Определите количество скрытых слоев от 1 до 5, количество нейронов в каждом из слоев до нескольких сотен, а также их функции активации (кусочно-линейная, сигмоидная, гиперболический тангенс и т.д.).

Для реализации нейронной сети была выбрана библиотека tensorflow 1.14.

В качестве алгоритма оптимизации был выбран “стохастический градиентный спуск” (SGD).

В качестве функций активации для скрытых слоёв была выбрана функция «Relu» (кусочно-линейная):

$$f(x) = \max(0, x) .$$

Для выходного слоя была выбрана функция активации softmax:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

Функция потерь была выбрана «категориальная перекрестная энтропия»:

$$L(\hat{y}, y) = - \sum_i y_{[i]} \log (\hat{y}_{[i]})$$

В ходе выполнения лабораторной работы было проведено 3 эксперимента с разной конфигурацией сети. Архитектура сети первого эксперимента представлена в таблице 1.

Таблица 1 – Архитектура сети первого эксперимента.

Слой		Размер	Ф-ция активации
Входной	-	784	-
1	FC	150	ReLU
2	FC	150	ReLU
Выходной	-	10	Softmax

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 100;
- batch size - 512.

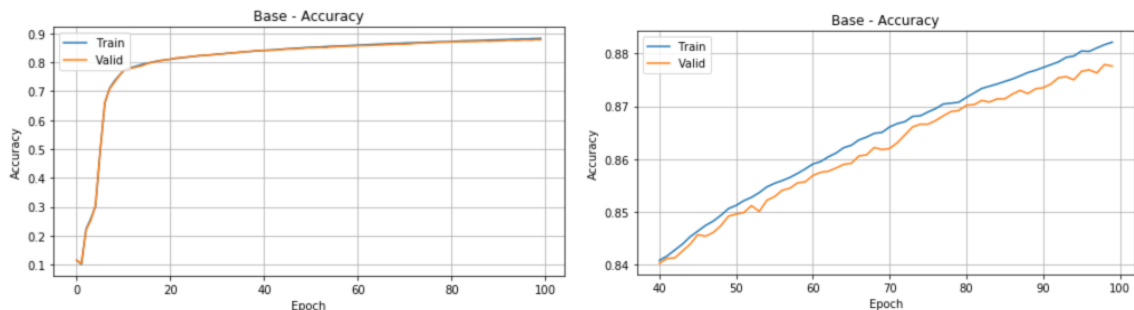


Рисунок 1 – график изменения ассурасу базовой модели.

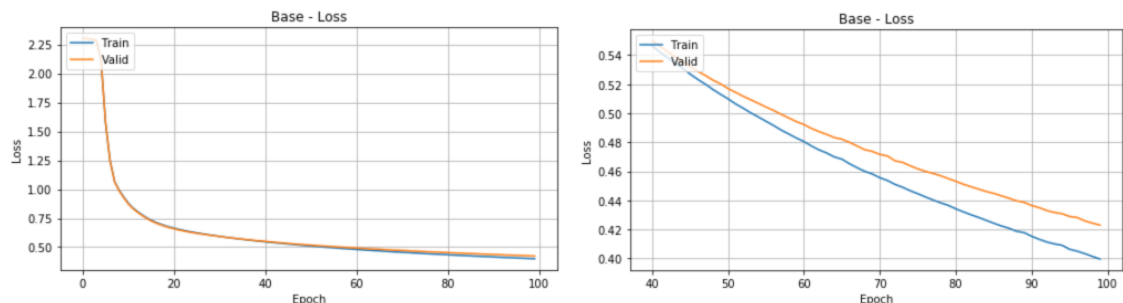


Рисунок 2 – график изменения loss базовой модели.

На тестовой выборке модель показала следующий результат:

- loss - 0.21200162;
- accuracy - 0.93979603.

2. Как улучшилась точность классификатора по сравнению с логистической регрессией?

Таблица 2 – Сравнение моделей.

Классификатор	Результат
Лог-регрессия	88.7%
Полносвязная нейронная сеть	93.97%

Из таблицы 2 видно, что полносвязная нейронная сеть значительно превосходит логистическую регрессию по точности, разница ~ 5%.

3. Используйте регуляризацию и метод сброса нейронов (dropout) для борьбы с переобучением. Как улучшилось качество классификации?.

Вторая модель была обучена с параметрами:

- epochs – 100;
- batch size - 512;
- L2 – 0.01;
- Dropout – 0.1.

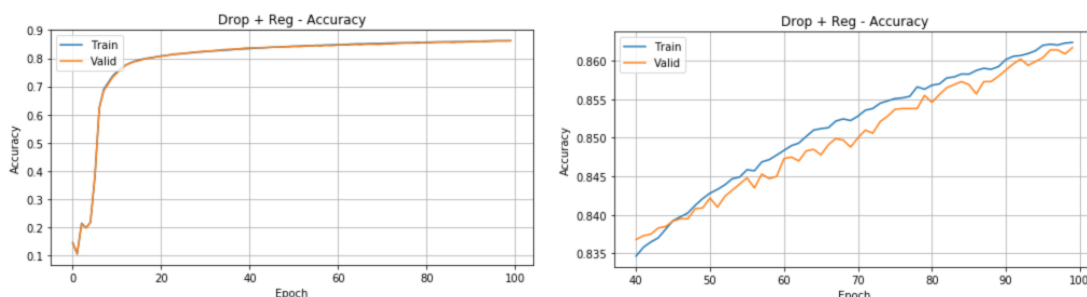


Рисунок 3 – график изменения ассигуры модели с регуляризацией и дропаутом.

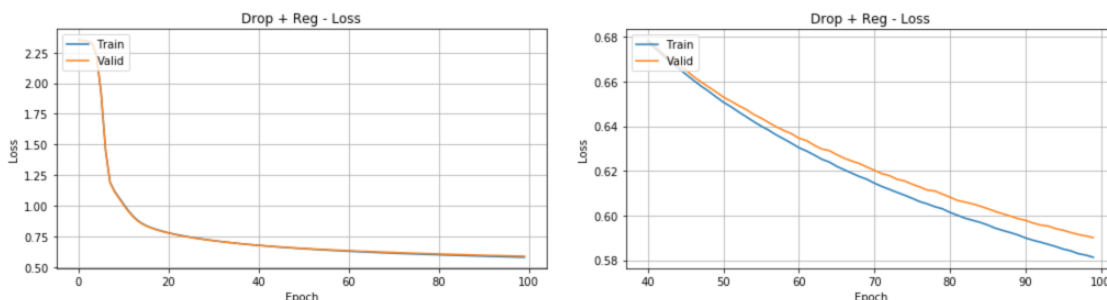


Рисунок 4 – график изменения loss модели с регуляризацией и дропаутом.

На тестовой выборке модель показала следующий результат:

- loss - 0.3661864;
- accuracy - 0.9312973.

Применение дропаута и регуляризации чуть-чуть снизило точность модели, но в тоже время снизилась разница между метриками на тренировочной и валидационной выборках, что свидетельствует об успешной борьбе с переобучением.

4. Воспользуйтесь динамически изменяемой скоростью обучения (learning rate). Наилучшая точность, достигнутая с помощью данной модели составляет 97.1%. Какую точность демонстрирует Ваша реализованная модель?

Третья модель обучалась с регуляризацией, дропаутом и адаптивным шагом и параметрами:

- epochs – 100;
- batch size - 512;
- L2 – 0.01;
- Dropout – 0.1.

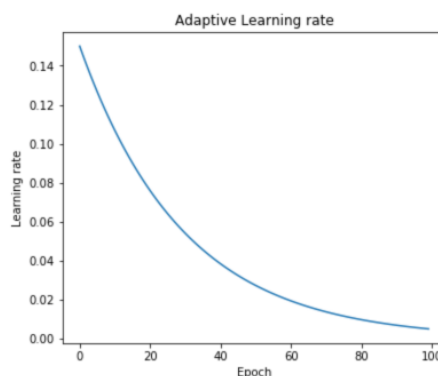


Рисунок 5 – график значения learning rate во время обучения.

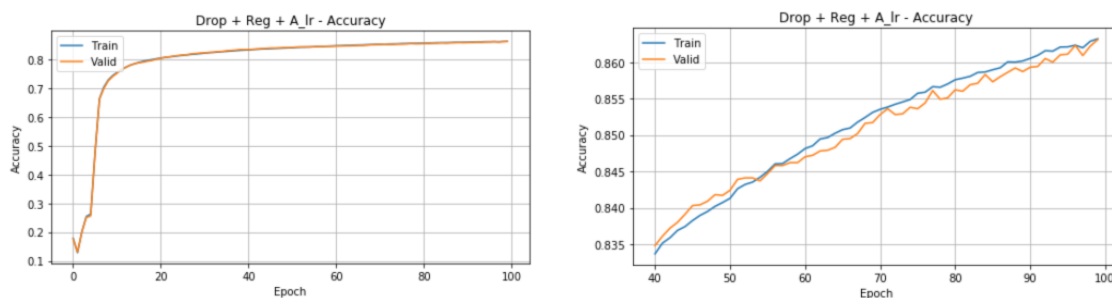


Рисунок 6 – график изменения ассигасу модели с адаптивным шагом.

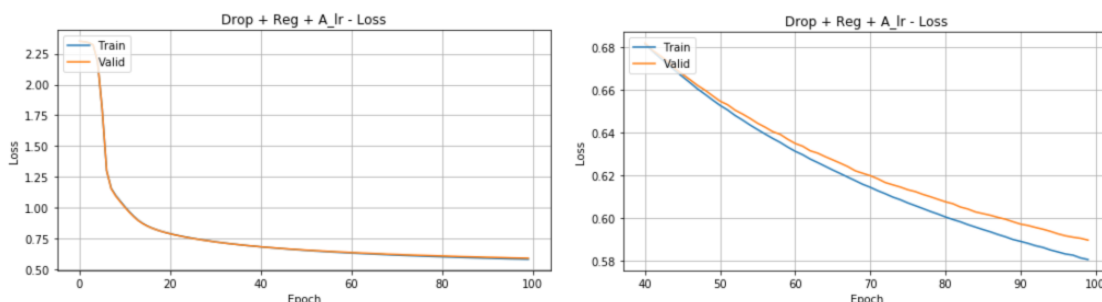


Рисунок 7 – график изменения loss модели с адаптивным шагом.

На тестовой выборке модель показала следующий результат:

- loss - 0.3660412;
- accuracy - 0.93031037.

Точность модели чуть-чуть ухудшилась.

Четвертая модель обучалась только с адаптивным шагом и параметрами:

- epochs – 100;
- batch size - 512.

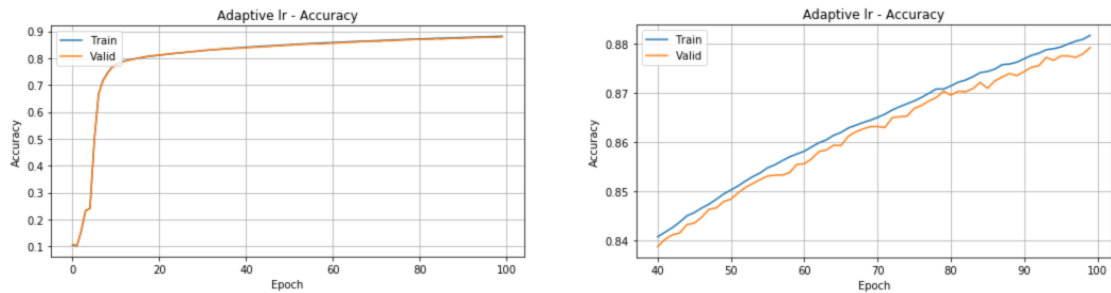


Рисунок 8 – график изменения ассигасу модели только с адаптивным шагом.

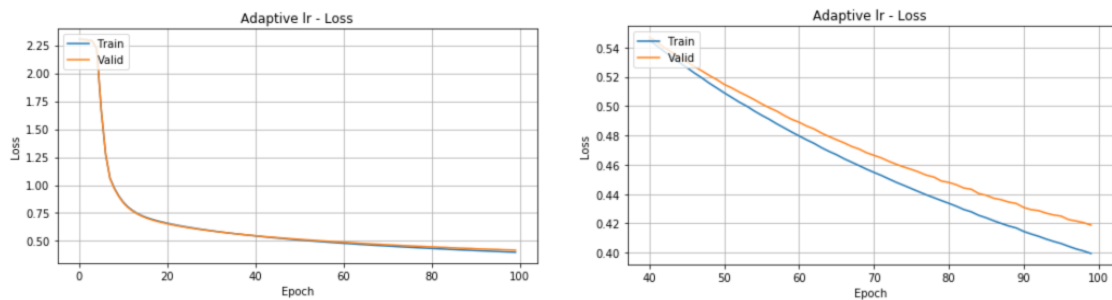


Рисунок 9 – график изменения loss модели только с адаптивным шагом.

На тестовой выборке модель показала следующий результат:

- loss - 0.21289702;
- accuracy - 0.9410023.

В первом эксперименте модель с адаптивным шагом показала наилучшую производительность.

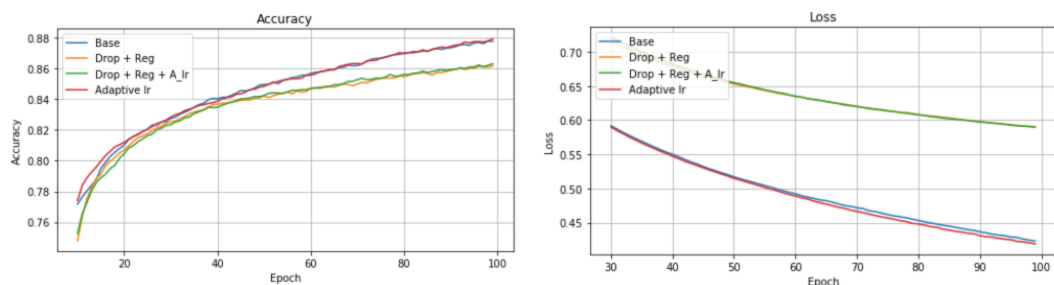


Рисунок 10 – график сравнения метрик обучения моделей.

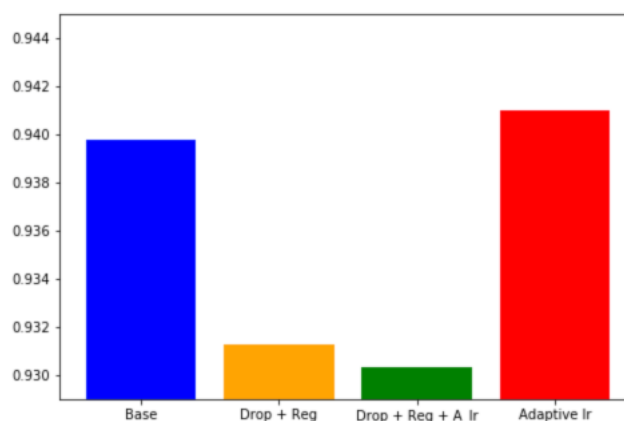


Рисунок 11 – график сравнения точности моделей первого эксперимента.

Для второго эксперимента были выбраны следующие параметры:

- epochs – 300;
- batch size - 512.

Архитектура сети осталась прежней.

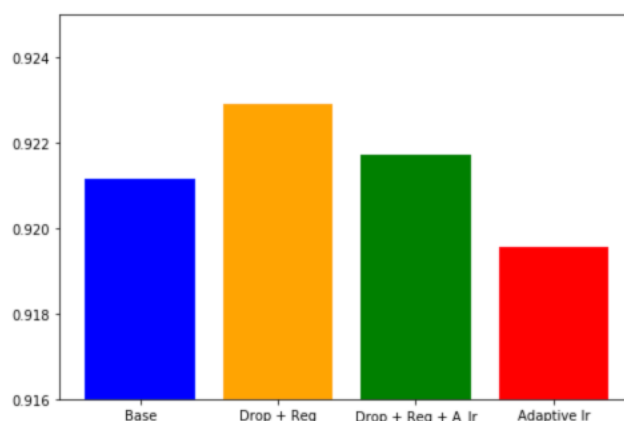


Рисунок 12 – график сравнения точности моделей второго эксперимента.

Если посмотреть на рисунок 12, то видно, что при увеличении количества эпох, самая большая точность у модели с дропаутом и регуляризацией, так как при большом кол-ве эпох модель начинает сильно переобучаться.

Для третьего эксперимента были выбраны следующие параметры:

- epochs – 150;
- batch size - 256.

Так же изменилась архитектура сети, новая архитектура описана в таблице 3.

Таблица 3 – Архитектура сети второго эксперимента.

Слой		Размер	Ф-ция активации
Входной	-	784	-
1	FC	150	ReLU
2	FC	120	ReLU
Выходной	-	10	Softmax

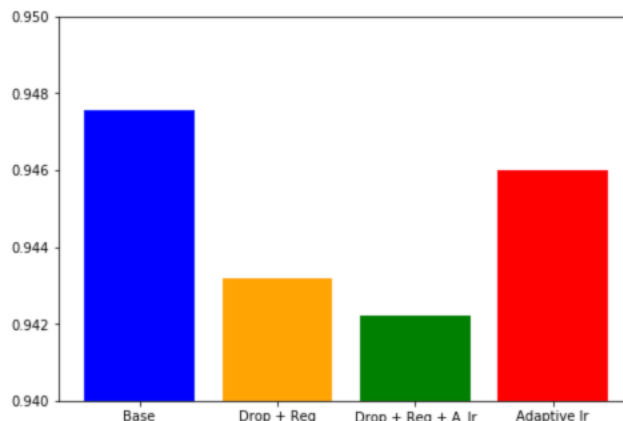


Рисунок 13 – график сравнения точности моделей второго эксперимента.

Наибольшая точность получилась в третьем эксперименте, причем у базовой модели, она составила 94.75%. Базовая модель скорее всего дала большую точность из за снижения сложности сети и уменьшения кол-ва эпох.

Вывод.

В ходе выполнения лабораторной работы я построил несколько моделей полносвязанных сетей для классификации изображений notMNIST. Так же для борьбы с переобучением, я применил методы регуляризации и дропаута. Для повышения скорости обучения я применил метод адаптивного шага обучения. Так же я дополнительно провел 3 эксперимента с разными гипер-параметрами и с разными архитектурами сетей, а результаты и сравнение отобразил в отчете.