

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных сетей и систем

Кафедра Информатики

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

по курсу «Машинное обучение»

Реализация глубокой нейронной сети

Студент:
гр. 758641
Ярош Г.И.

Проверил:
Заливако С. С.

Минск, 2019

СОДЕРЖАНИЕ

РЕАЛИЗАЦИЯ ГЛУБОКОЙ НЕЙРОННОЙ СЕТИ	3
1. Цель	3
2. Полносвязная нейронная сеть	3
3. Применение регуляризации и метода сброса нейронов	5
4. Применение адаптивного шага обучения.....	7
5. Сравнение построенных моделей	8
6. Анализ полученных результатов.....	9
7. Вывод	11
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	12

РЕАЛИЗАЦИЯ ГЛУБОКОЙ НЕЙРОННОЙ СЕТИ

1. Цель

Реализовать полносвязную нейронную сеть для решения задачи классификации изображений из набора данных notMNIST. Провести исследование эффективности использования метода сброса нейронов, регуляризации и адаптивного шага обучения для обучения нейронной сети.

2. Полносвязная нейронная сеть

В данной лабораторной работе проводилось исследование возможности классификации изображений из набора данных notMNIST с использованием полносвязных нейронных сетей.

Для обучения полносвязной нейронной сети, исходные изображения были преобразованы в векторное представление. В качестве выходов использовался массив векторов длиной 10, где i -ая компонента вектора равна 1, если соответствующее данному вектору изображение принадлежит к классу i , а остальные компоненты равны 0.

Построенная мною полносвязная сеть состоит из 2 скрытых слоев, каждый из которых содержит 256 нейронов с кусочно-линейной функцией активации. Выходной слой содержит 10 нейронов и использует функцию активации softmax (рисунок 1).

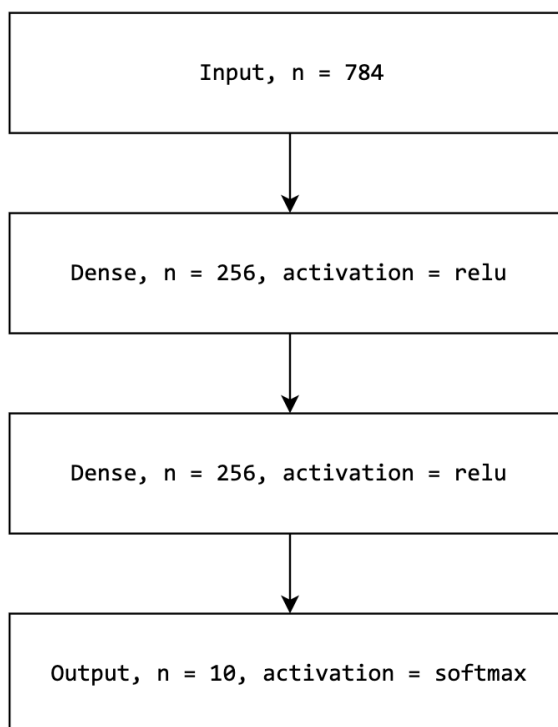


Рис. 1. Построенная полносвязная сеть.

В качестве функции потерь мною была использована категориальная кроссэнтропия, вычисляемая по следующей формуле:

$$CE = - \sum_{i=1}^C y_i^0 \log y_i,$$

где C – количество классов, y^0 – исходный выходной вектор, y – выход нейронной сети с примененной функцией softmax.

Точность классификации вычислялась по следующей формуле:

$$accuracy = \frac{1}{N} \sum_{i=1}^N 1 \left(\max_j(y_j) = \max_j(y_j^0) \right),$$

где N – размер тестовой выборки, \max_j – индекс элемента вектора с максимальным значением.

В качестве алгоритма оптимизации был использован стохастический градиентный спуск с коэффициентом обучения $LR = 0.001$.

Перед обучением нейронной сети из обучающей выборки была выделена валидационная размером 10% от исходной. На каждой эпохе обучения обучающая выборка разбивалась на батчи размером в 1000 элементов, на каждом из которых проводилась итерация обучения. После обучения на всех батчах для валидационной выборки вычислялась функция потерь и точность. Если функция потерь на валидационной выборке не уменьшалась в течении 20 эпох, то обучение останавливалось, а состояние весов модели возвращалось в наилучшему.

Зависимости значения функции потерь и точности для обучающей и валидационной выборок представлены на рисунках 2 и 3.

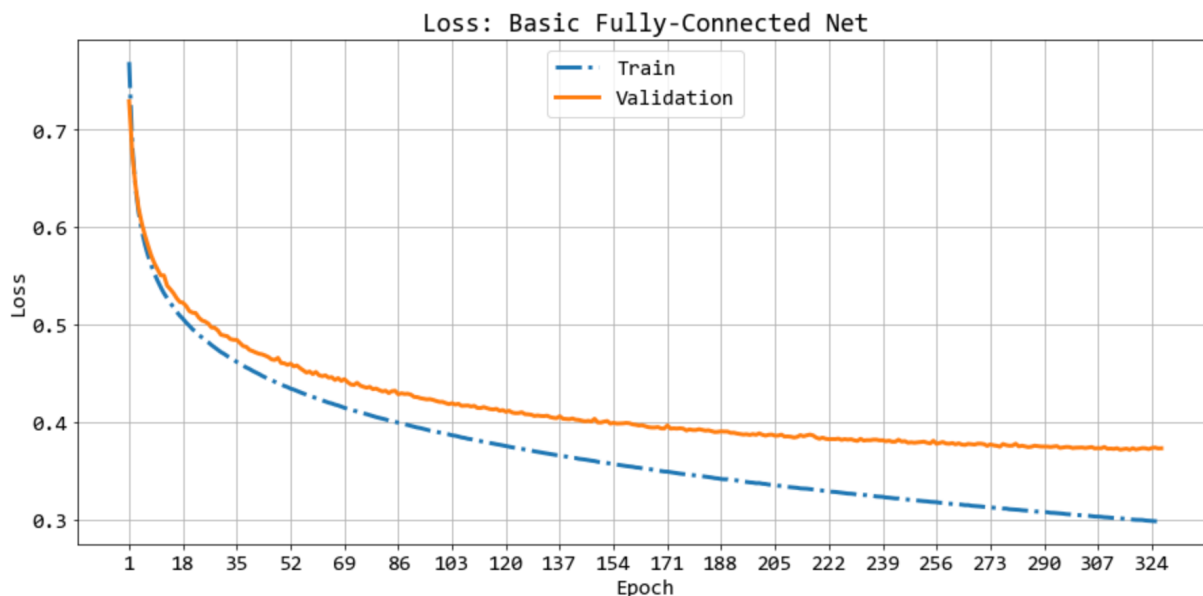


Рис. 2. Уменьшение значения функции потерь при обучении.

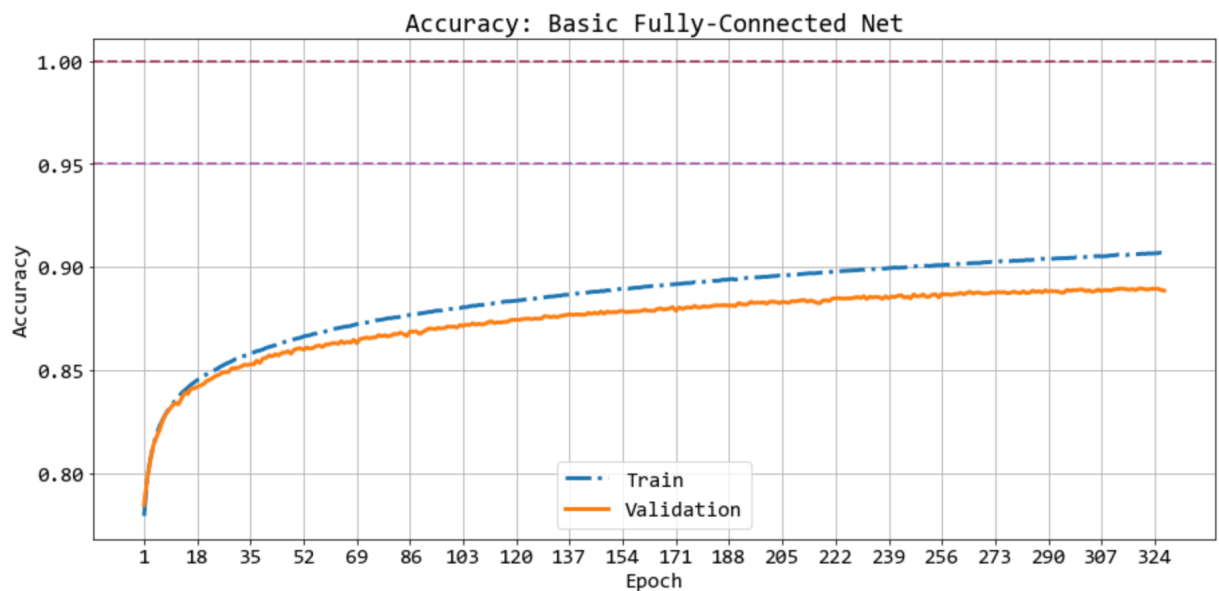


Рис. 3. Увеличение точности классификации при обучении.

В результате, после 277 эпох, обучение нейронной сети остановилось. Точность классификации на тестовой выборке составила 0.945.

3. Применение регуляризации и метода сброса нейронов

При обучении модели неизбежно возникает эффект переобучения. Для борьбы с ним используется метод сброса нейронов и добавление регуляризации к функции потерь.

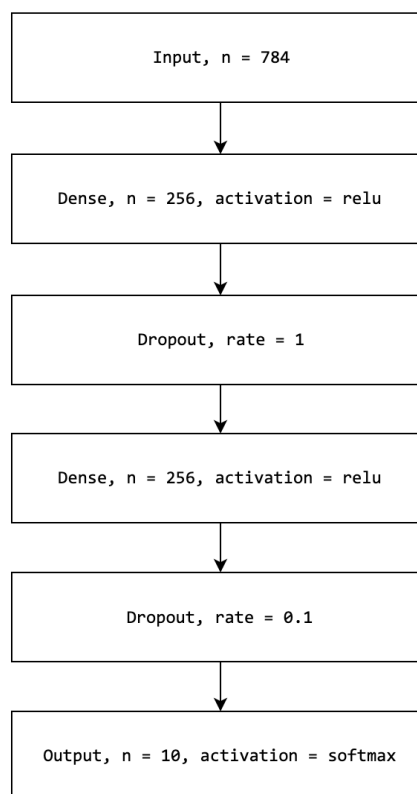


Рис. 4. Структура нейронной сети с слоями сброса нейронов.

Построенная полносвязная сеть была модифицирована следующим образом: после каждого внутреннего слоя добавлен слой сброса нейронов с процентом сброса равным 10. В функцию потерь было добавлено дополнительное слагаемое, равное сумме L2 норм от весов каждого слоя модели. Схема получившейся нейронной сети представлена на рисунке 4.

Обучение модели происходило таким же образом, как и в предыдущем пункте. Изменения функции потерь и точности при обучении отображены на рисунках 5, 6.

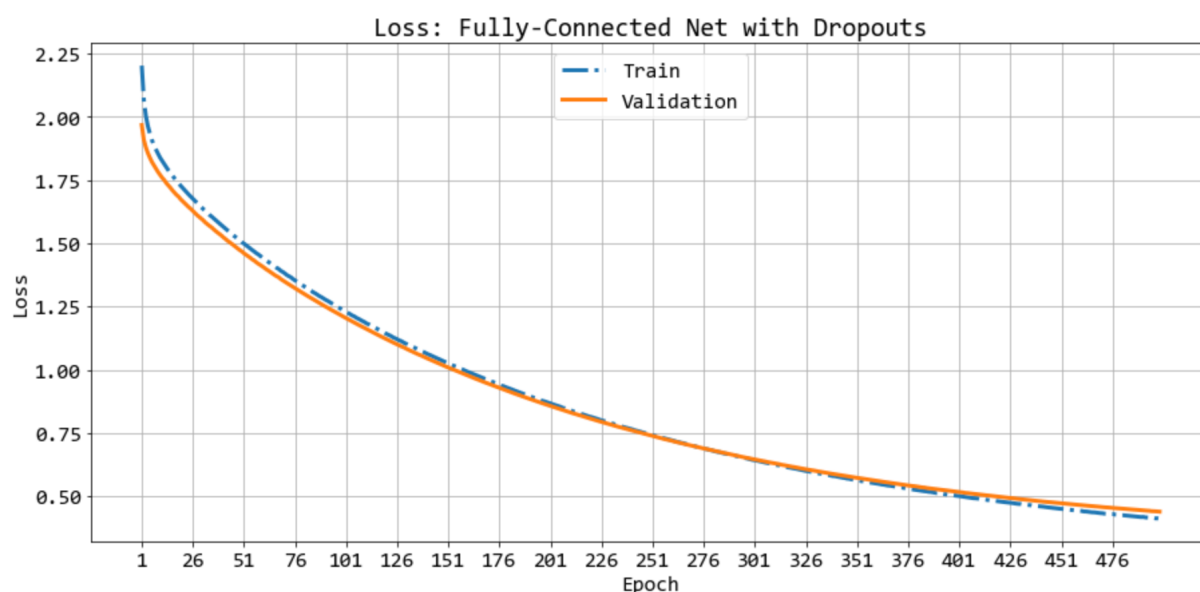


Рис. 5. Изменение значения функции потерь в процессе обучения.

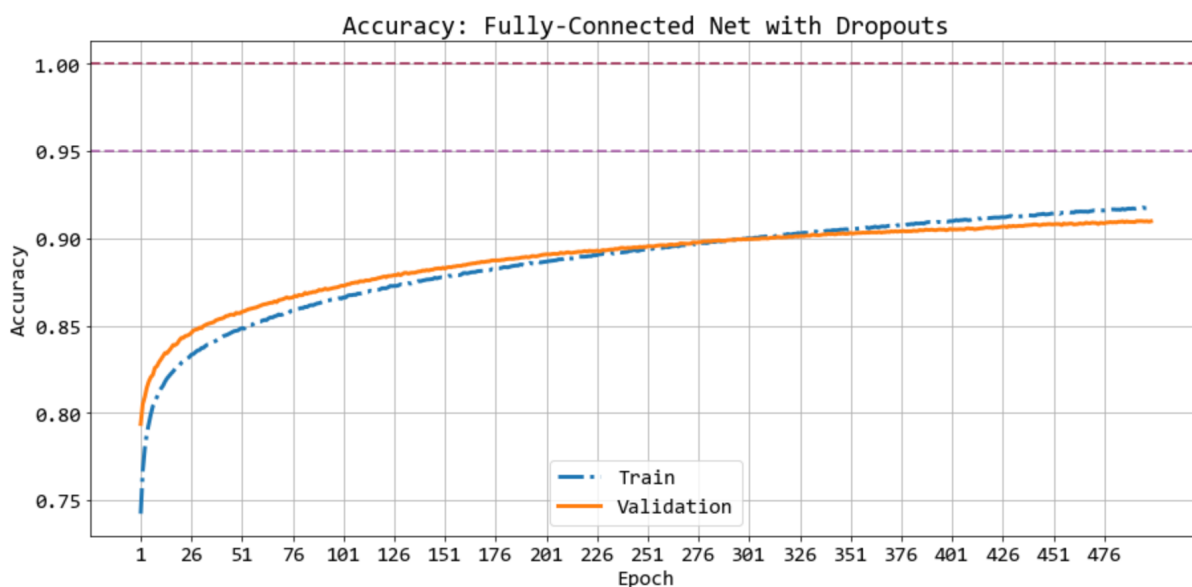


Рис. 6. Изменение точности классификации в процесса обучения.

Обучение длилось 500 эпох. Точность классификации на тестовой выборке достигла 0.96.

4. Применение адаптивного шага обучения

Для ускорения процесса обучения применяется метод адаптивного шага обучения. В данном случае был применен другой метод оптимизации с адаптивным шагом обучения Adam.

Изменения функции потерь и точности классификации при обучении приведены на рисунках 7,8.

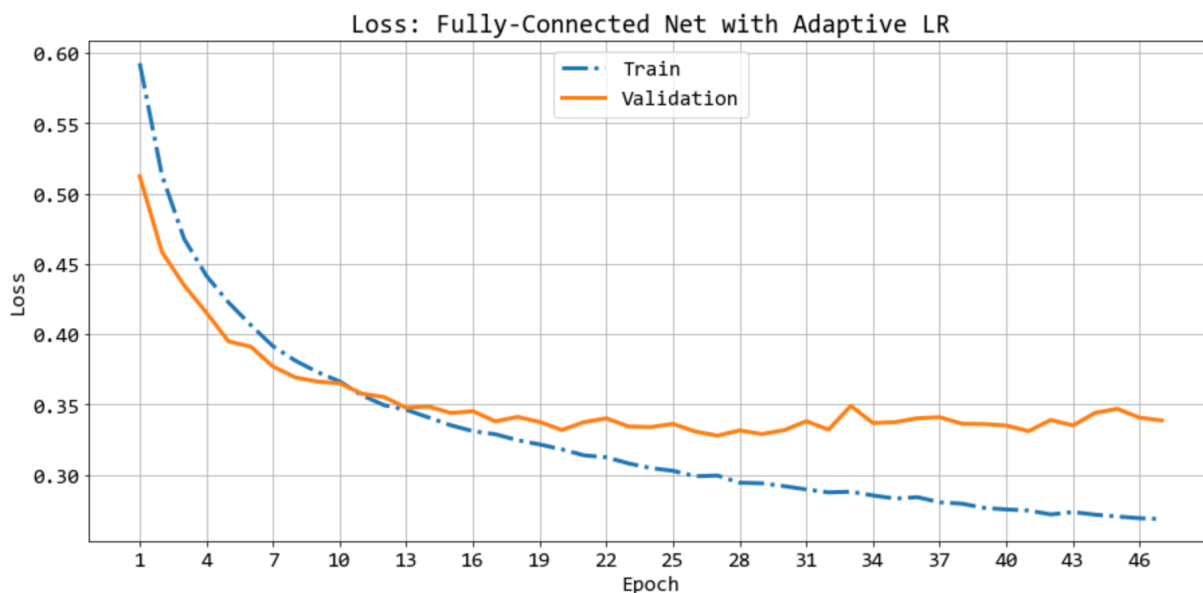


Рис. 7. Изменение значения функции потерь при обучении.

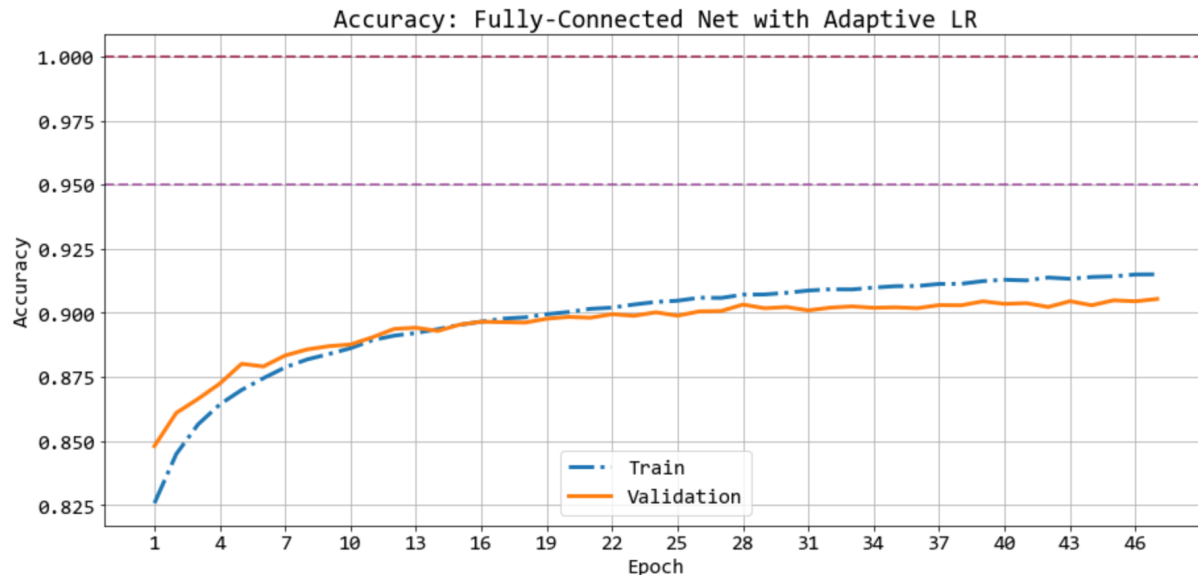


Рис. 8. Изменение точности классификации в процессе обучения.

Обучение нейронной сети заняло 48 эпох. Точность классификации на тестовой выборке составила 0.954.

5. Сравнение построенных моделей

В процессе работы было построено три нейронные сети с одинаковой структурой, но с разными параметрами и алгоритмом оптимизации.

Сравнение изменения функции потерь и точности классификации при обучении для каждой из моделей приведено на рисунках 9, 10.

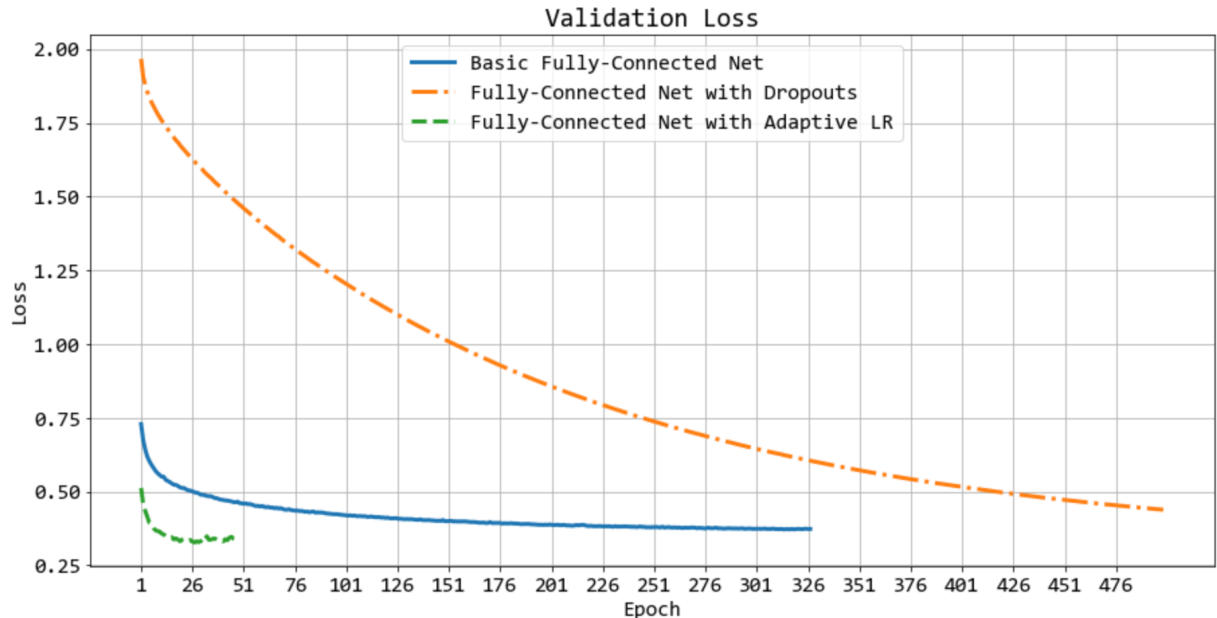


Рис. 9. Изменение функции потерь в процессе обучения.

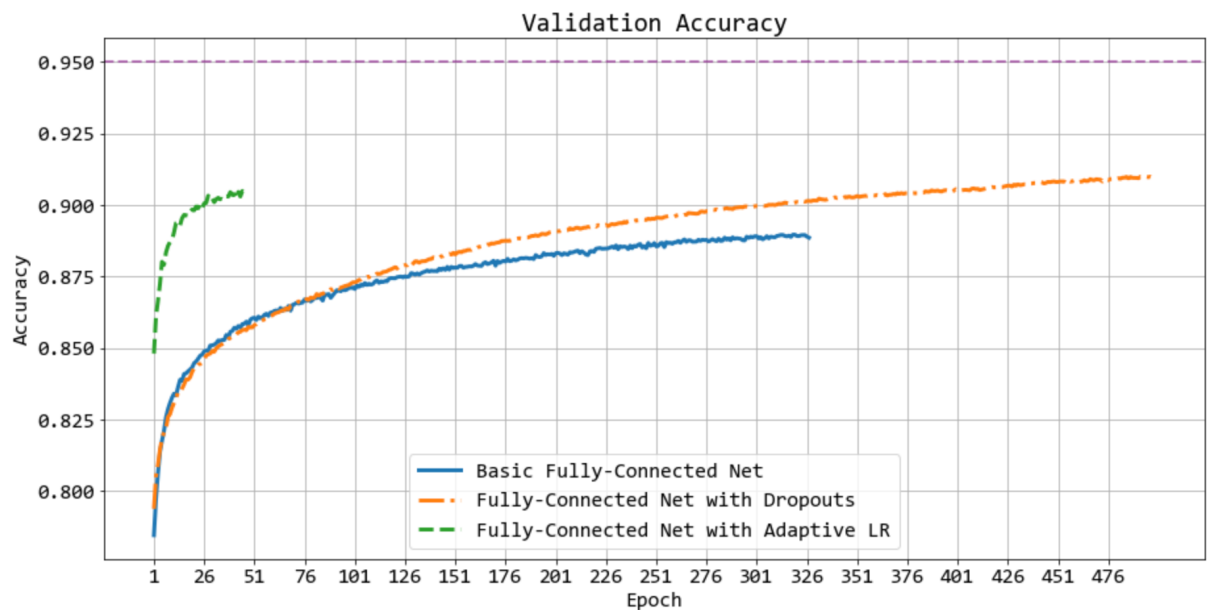


Рис. 10. Изменение точности классификации в процессе обучения.

Как видно из приведенных зависимостей, модель с адаптивным шагом обучилась быстрее всего. Модель со сбросом нейронов и регуляризацией обучалась наибольшее количество эпох, что связано с наступлением эффекта переобучения.

Сравнения достигнутых точностей моделей на тестовой выборке приведено на рисунке 11.

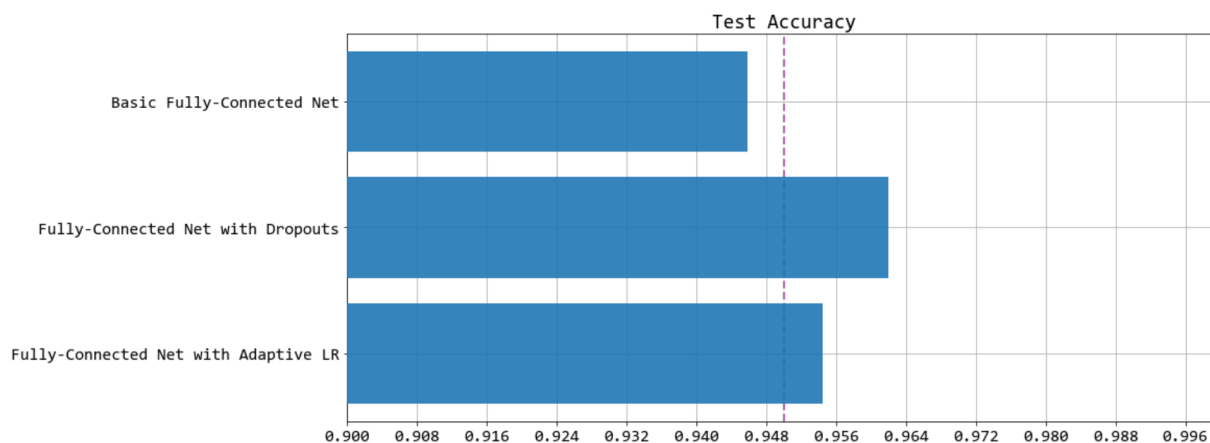


Рис. 11. Сравнение достигнутых точностей классификации.

На рисунке видно, что максимальная точность классификации достигнута во второй модели со сбросом нейронов и регуляризацией. Для модели точность классификации чуть ниже, что может быть обусловлено малым количеством эпох обучения. Наименьшая точность у простой модели, что объясняется быстрым наступлением эффекта переобучения.

6. Анализ полученных результатов

В процессе работы, все модели показывали точность на тестовой выборке значительно больше, чем на тренировочной в процессе обучения. Для исследования этого эффекта, мною был проанализирован исходный набор данных.

Оказалось, что тестовая и тренировочная выборки содержат много изображений, мало похожих на изображения букв. Также некоторые изображения содержат букву, которая воспринимается другой, нежели соответствующая метка. На рисунках 12 и 13 приведены примеры изображений, на которых обученная сеть ошиблась.

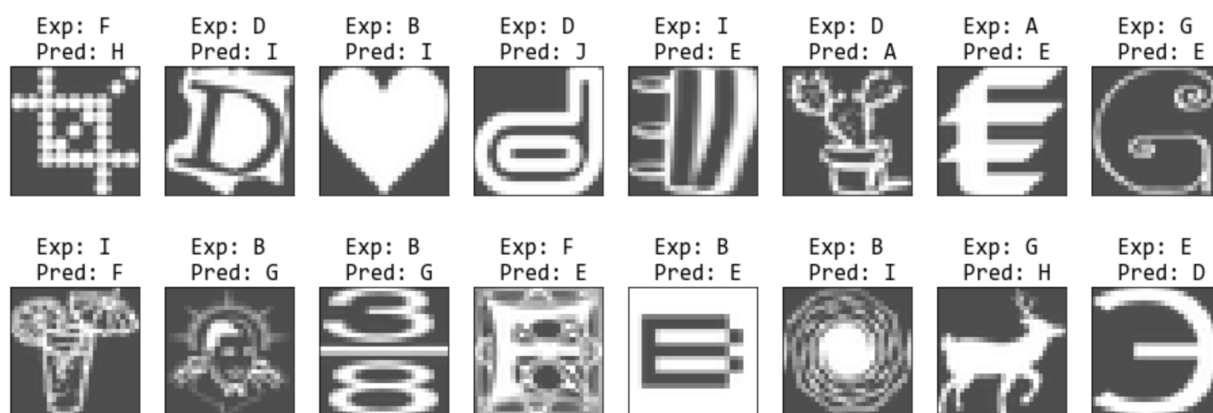


Рис. 12. Примеры изображений из тренировочной выборки, классифицированных неверно.

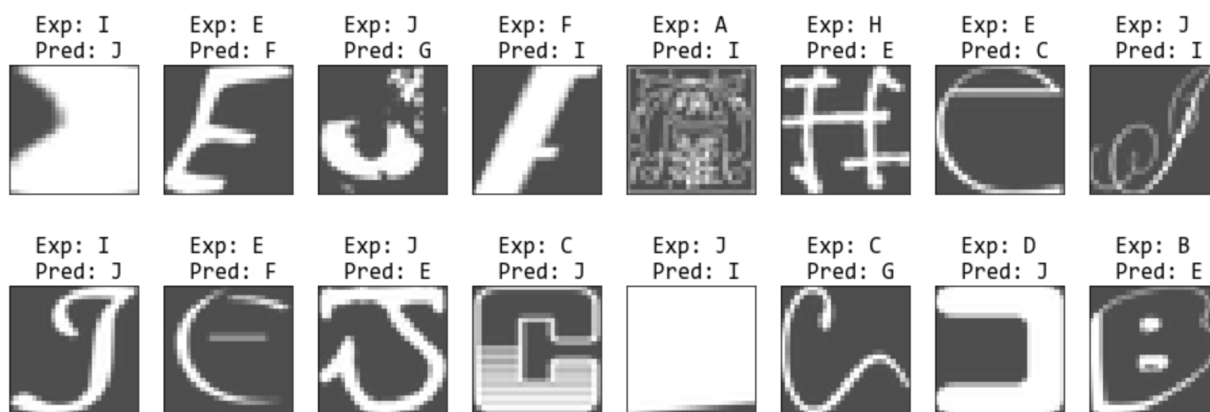


Рис. 13. Примеры изображений из тестовой выборки, классифицированные неправильно.

Эффект с большей точностью на тестовой выборке, чем на тренировочной можно объяснить большей представленностью «странных» изображений в тренировочной выборке.

Также было проведено исследование, какие классы наиболее сложны для классификации. Распределения изображений, классифицированных неверно, по каждому классу для тренировочной и тестовой выборок приведены на рисунках 14, 15.

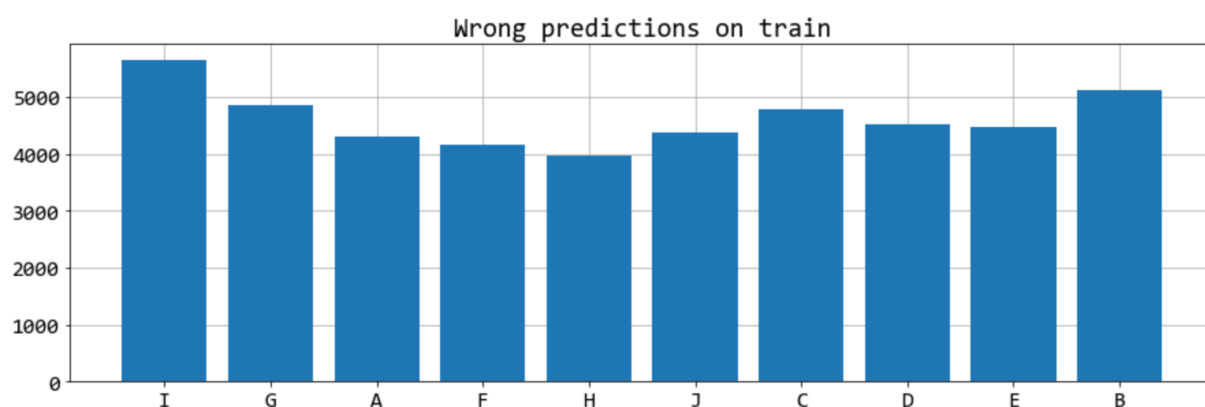


Рис. 14. Распределение неверно классифицированных изображений в тренировочной выборке.

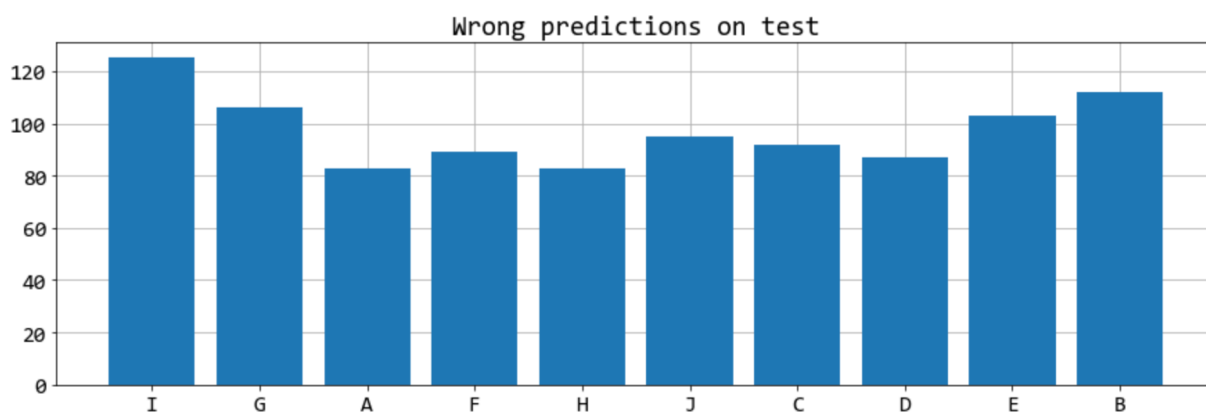


Рис. 15. Распределение неверно классифицированных изображений в тестовой выборке.

Из диаграмм видно, что есть некоторое различие в сложности классификации, однако оно незначительно. Также нет класса, явно распознаваемого наилучшим или наихудшим образом.

7. Вывод

В результате работы была построена модель полносвязной нейронной сети для классификации изображений notMNIST. Был применен метод сброса нейронов и регуляризация для борьбы с эффектом обчения. Был применен метод адаптивного шага обучения для уменьшения времени обучения. Было проведено сравнение и исследование полученных результатов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] – notMNIST dataset [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html>.

[2] – Cross Entropy Loss [Электронный ресурс]. – Электронные данные. – Режим доступа: https://gombru.github.io/2018/05/23/cross_entropy_loss/

[3] – Adam Optimizer [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf/train/AdamOptimizer

[4] – The APIs for neural networks in TensorFlow [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.oreilly.com/ideas/the-apis-for-neural-networks-in-tensorflow>