

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

|           |                             |
|-----------|-----------------------------|
| Факультет | Компьютерных сетей и систем |
| Кафедра   | Информатики                 |

МАШИННОЕ ОБУЧЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА №6  
«Применение сверточных нейронных сетей (многоклассовая  
классификация)»

БГУИР 1-40 81 04

Магистрант:  
гр. 858641  
Кукареко А.В.

Проверил:  
Стержанов М. В.

Минск, 2020

## ХОД РАБОТЫ

### Данные.

Набор данных для распознавания языка жестов, который состоит из изображений размерности 28x28 в оттенках серого (значение пикселя от 0 до 255). Каждое из изображений обозначает букву латинского алфавита, обозначенную с помощью жеста, как показано на рисунке ниже (рисунок цветной, а изображения в наборе данных в оттенках серого). Обучающая выборка включает в себя 27,455 изображений, а контрольная выборка содержит 7172 изображения. Данные в виде csv-файлов можно скачать на сайте Kaggle -> <https://www.kaggle.com/datamunge/sign-language-mnist> .

### Задание.

1. Загрузите данные. Разделите исходный набор данных на обучающую и валидационную выборки;
2. Реализуйте глубокую нейронную сеть со сверточными слоями. Какое качество классификации получено? Какая архитектура сети была использована?
3. Примените дополнение данных (data augmentation). Как это повлияло на качество классификатора?
4. Поэкспериментируйте с готовыми нейронными сетями (например, AlexNet, VGG16, Inception и т.п.), применив передаточное обучение. Как это повлияло на качество классификатора? Можно ли было обойтись без него? Какой максимальный результат удалось получить на контрольной выборке?

### Результат выполнения:

1. Загрузите данные. Разделите исходный набор данных на обучающую, валидационную и контрольную выборки.

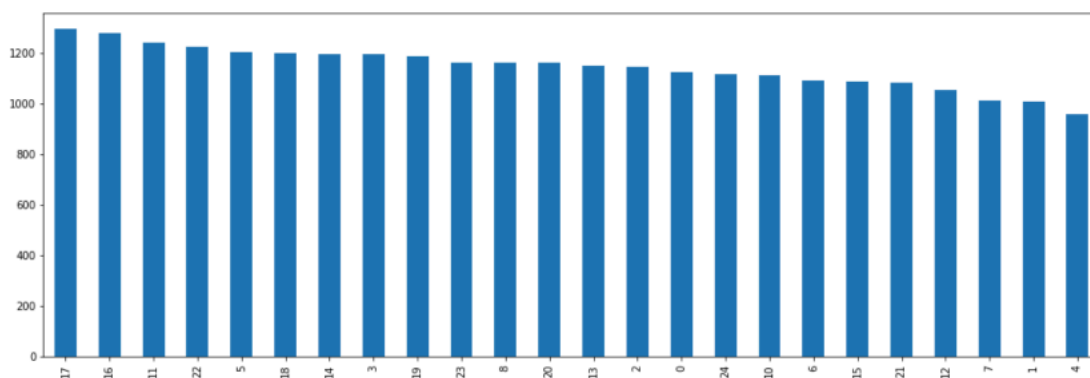


Рисунок 1 - кол-во изображений в каждом классе.

Как видно из рисунка 1 – данные в классах не совсем сбалансированы.

Классы можно сбалансировать с помощью аугментации, до размеров мажоритарного класса. Такая техника называется oversampling.

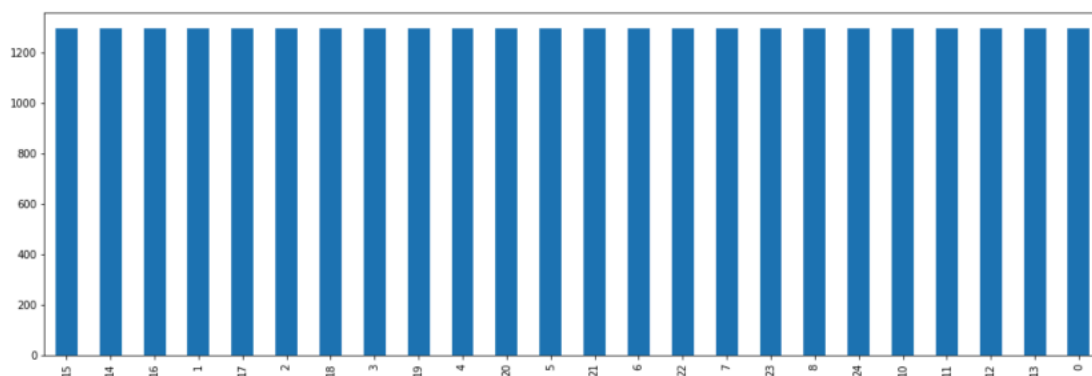


Рисунок 2 - кол-во изображений в каждом классе после oversampling.

Теперь мы можем применять метрику ассигасы для определения точности классификатора.

Всего изображений 38 228. Они были разделены на:

- тренировочную выборку – 23 292 изображений;
- валидационную выборку – 7 764 изображений;
- тестовую выборку – 7 172 изображений.

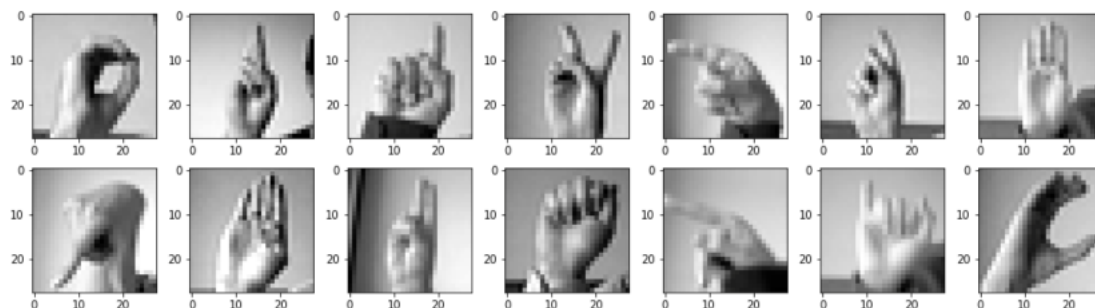


Рисунок 3 – пример данных из набора «sign-language-mnist».

2. Реализуйте глубокую нейронную сеть с как минимум тремя сверточными слоями. Какое качество классификации получено?

Для реализации нейронной сети была выбрана библиотека tensorflow 1.14.

Архитектура нейронной сети представлена в таблице 1.

Таблица 1 – Архитектура нейронной сети.

| Слой    |          | Размер  | Фильтры | Ядро  | Смещение | Активация |
|---------|----------|---------|---------|-------|----------|-----------|
| Входной | -        | 28x28x1 | -       | -     | -        | -         |
| 1       | Conv2d   | 28x28   | 32      | 4 x 4 | 1        | ReLU      |
| 2       | B. Norm. | -       | -       | -     | -        | -         |
| 3       | Max Pool | 14x14   | 32      | 2 x 2 | 2        | -         |
| 4       | Conv2d   | 14x14   | 64      | 4 x 4 | 1        | ReLU      |
| 5       | B. Norm. | -       | -       | -     | -        | -         |

|          |          |      |     |       |   |         |
|----------|----------|------|-----|-------|---|---------|
| 6        | Max Pool | 7x7  | 64  | 2 x 2 | 2 | -       |
| 7        | Conv2d   | 7x7  | 128 | 4 x 4 | 1 | ReLU    |
| 8        | B. Norm. | -    | -   | -     | - | -       |
| 9        | Max Pool | 4x4  | 128 | 2 x 2 | 2 | -       |
| 10       | Flatten  | 2048 | -   | -     | - | -       |
| 11       | FC       | 128  | -   | -     | - | ReLU    |
| 12       | Dropout  |      |     |       |   |         |
| 13       | FC       | 128  | -   | -     | - | ReLU    |
| 14       | Dropout  |      |     |       |   |         |
| Выходной | FC       | 24   | -   | -     | - | Softmax |

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 30;
- batch size - 64;
- dropout - 0.42.

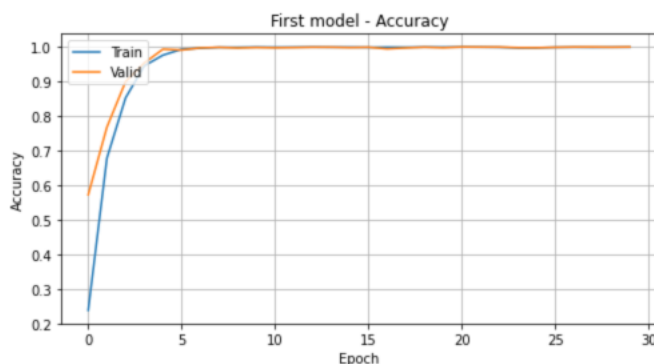


Рисунок 4 – график изменения ассигура первой модели.

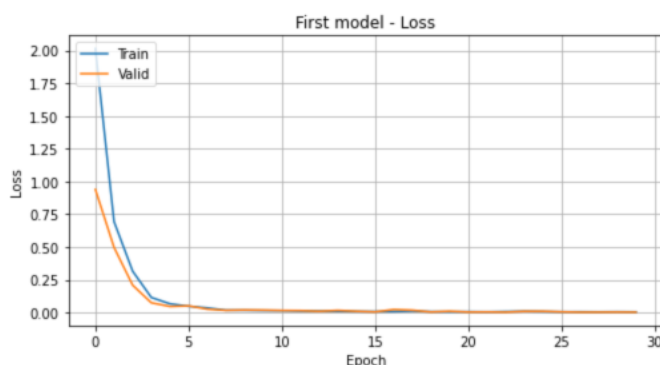


Рисунок 5 – график изменения loss первой модели.

На тестовой выборке модель показала следующий результат:

- loss - 0.1423;
- accuracy – 0.9872.

3. Примените дополнение данных (data augmentation). Как это повлияло на качество классификатора?

Для «data augmentation» была использована библиотека “keras ImageDataGenerator”.

Для генерации картинок использовались следующие параметры:

- rotation\_range - 40;
- width\_shift\_range - 0.2;
- height\_shift\_range - 0.2;
- zoom\_range - 0.2;
- horizontal\_flip- True.

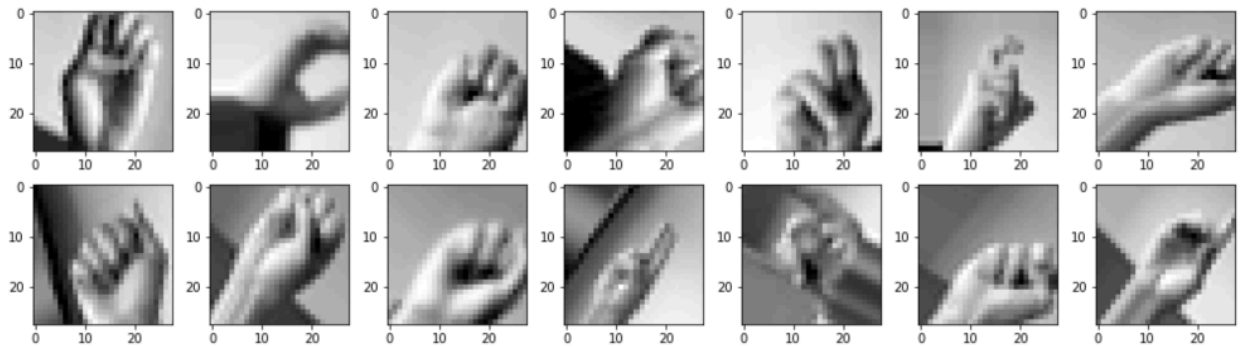


Рисунок 6 – пример аугментированных изображений в итерации 1.

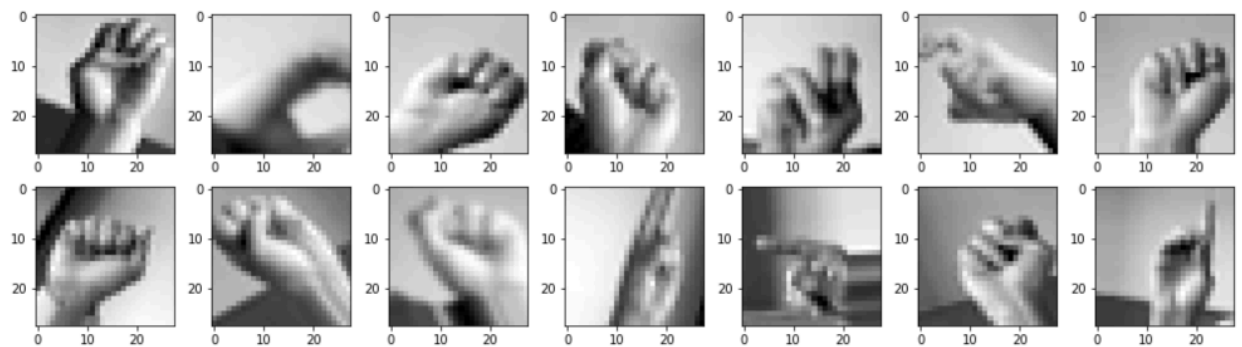


Рисунок 7 – пример аугментированных изображений в итерации 1.

Если сравнить рисунки 5 и 6 можно увидеть, как «ImageDataGenerator» преобразует одни и те же изображения.

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 30;
- batch size - 64;
- dropout - 0.2.

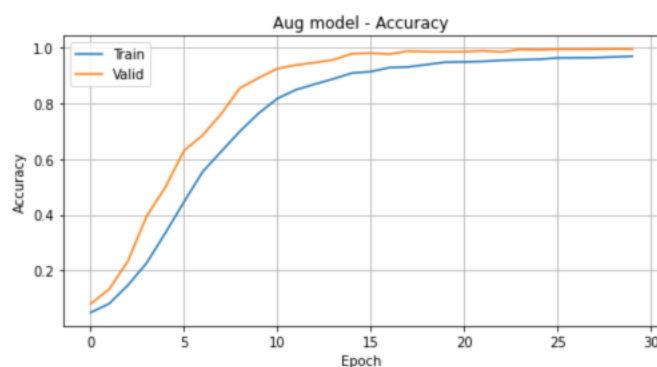


Рисунок 8 – график изменения ассигасу модели с применением аугментированных данных.

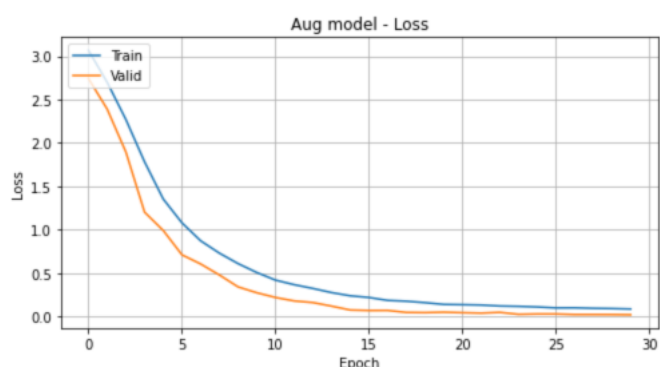


Рисунок 9 – график изменения loss первой модели с применением аугментированных данных.

На тестовой выборке модель показала следующий результат:

- loss - 0.0402;
- accuracy - 0.9906.

Если посмотреть на рисунки 7 и 8, можно увидеть, что даже при меньшем значении dropout, переобучение модели отсутствует.

4. Поэкспериментируйте с готовыми нейронными сетями (например, AlexNet, VGG16, Inception и т.п.), применив передаточное обучение. Как это повлияло на качество классификатора? Какой максимальный результат удалось получить на сайте Kaggle? Почему?

Для передаточного обучения была выбрана сеть VGG16. Готовые сети с весами предоставляет библиотека «keras.applications». Архитектуру сети VGG16 можно посмотреть на рисунке 9.

|        | Layer           | Feature Map | Size            | Kernel Size | Stride | Activation |
|--------|-----------------|-------------|-----------------|-------------|--------|------------|
| Input  | Image           | 1           | 224 x 224 x 3   | -           | -      | -          |
| 1      | 2 X Convolution | 64          | 224 x 224 x 64  | 3x3         | 1      | relu       |
|        | Max Pooling     | 64          | 112 x 112 x 64  | 3x3         | 2      | relu       |
| 3      | 2 X Convolution | 128         | 112 x 112 x 128 | 3x3         | 1      | relu       |
|        | Max Pooling     | 128         | 56 x 56 x 128   | 3x3         | 2      | relu       |
| 5      | 2 X Convolution | 256         | 56 x 56 x 256   | 3x3         | 1      | relu       |
|        | Max Pooling     | 256         | 28 x 28 x 256   | 3x3         | 2      | relu       |
| 7      | 3 X Convolution | 512         | 28 x 28 x 512   | 3x3         | 1      | relu       |
|        | Max Pooling     | 512         | 14 x 14 x 512   | 3x3         | 2      | relu       |
| 10     | 3 X Convolution | 512         | 14 x 14 x 512   | 3x3         | 1      | relu       |
|        | Max Pooling     | 512         | 7 x 7 x 512     | 3x3         | 2      | relu       |
| 13     | FC              | -           | 25088           | -           | -      | relu       |
| 14     | FC              | -           | 4096            | -           | -      | relu       |
| 15     | FC              | -           | 4096            | -           | -      | relu       |
| Output | FC              | -           | 1000            | -           | -      | Softmax    |

Рисунок 10 – архитектура сети vgg16.

У сети VGG16 были убраны последние 4 полносвязных слоя и добавлены 3 новых. Архитектуру новой модели можно увидеть в таблице 2.

Таблица 2 – Архитектура новой модели с применением передаточного обучения.

| Слой     |         | Размер  | Активация |
|----------|---------|---------|-----------|
| Входной  | -       | 32x32x3 | -         |
| -        | VGG16   | 1x1x512 | -         |
| -        | Flatten | 512     | -         |
| -        | FC      | 128     | ReLU      |
| -        | Dropout |         |           |
| -        | FC      | 128     | ReLU      |
| -        | Dropout |         |           |
| Выходной | FC      | 24      | Softmax   |

Так же было проведено 2 эксперимента:

- Модель была обучена на оригинальном наборе данных.
- Модель была обучена на аугментированном наборе данных

Так как минимальным входным размером изображений сети VGG16 является размер: 32x32x3, то все изображения пришлось модифицировать.

- по краям добавился паддинг в размере 4 пикселей;
- 1 канал продублировался на 3.

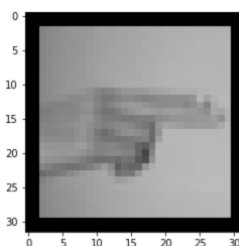


Рисунок 11 – пример модифицированного изображения.

Обучение моделей запускалось со следующими параметрами:

- epochs – 20;
- batch size - 64;
- dropout - 0.5.

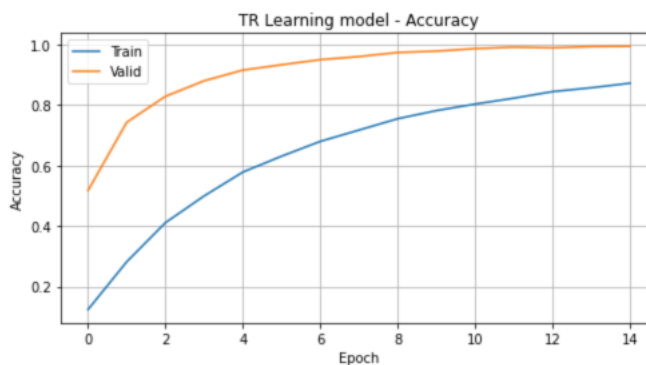


Рисунок 12 – график изменения ассигасу модели с применением передаточного обучения данных (без аугментации).

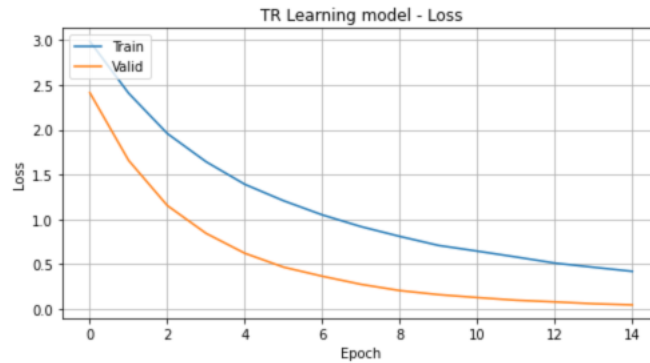


Рисунок 13 – график изменения loss первой модели с применением передаточного обучения данных (без аугментации).

На тестовой выборке модель показала следующий результат:

- loss - 0.1247;
- accuracy - 0.9639.

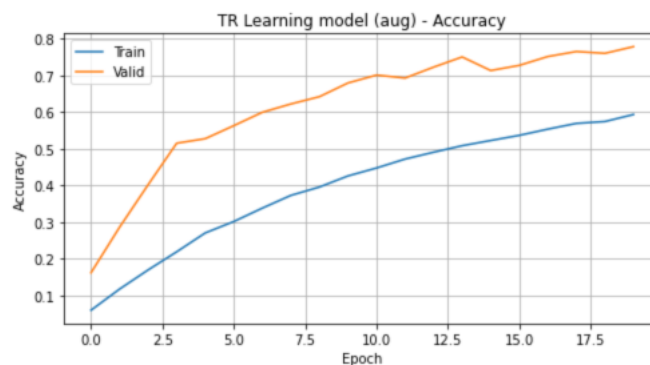


Рисунок 14 – график изменения ассигасу модели с применением передаточного обучения данных (с аугментацией).

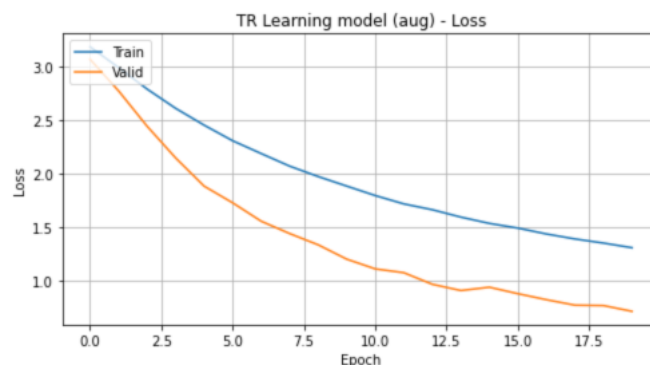


Рисунок 15 – график изменения loss первой модели с применением передаточного обучения данных (с аугментацией).

На тестовой выборке модель показала следующий результат:

- loss - 0.8956;
- accuracy - 0.7008.



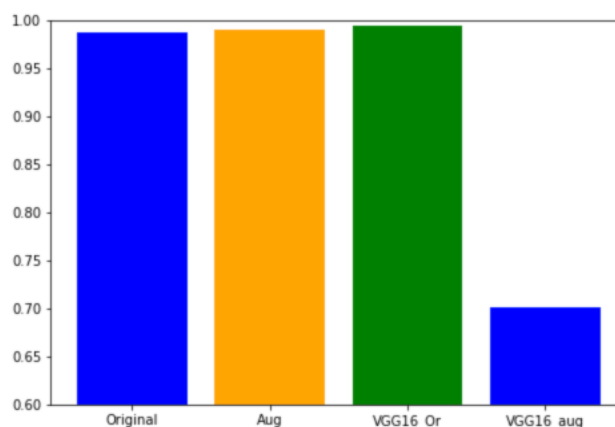


Рисунок 16 – сравнение результатов моделей.

Таблица 3 – сравнение результатов моделей на тестовой выборке.

| Модель                                   | Accuracy          |
|--|-------------------|
| Conv + augmented data                    | 0.9906 ( 99.06% ) |
| Conv + original data                     | 0.9872 ( 98.72% ) |
| Transfer learning vgg16 + original data  | 0.9639 ( 96.39% ) |
| Transfer learning vgg16 + augmented data | 0.7008 ( 70.08% ) |

## Вывод.

В ходе выполнения лабораторной работы я построил одну модель использующих сверточные слои для классификации изображений жестов. Обучил эту модель без применения аугментации, и с применением аугментации данных. Так же для решения данной задачи были использованы готовые модели, а именно VGG16 и техника передаточного обучения.

После обучения всех моделей и анализа результатов, можно сделать вывод, что техника «передаточного обучения» не во всех случаях дает прирост точности модели. По результатам этой работы видно, что модели использующие передаточное обучение показали худший результат.