

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

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

МАШИННОЕ ОБУЧЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА №5
«Применение сверточных нейронных сетей (бинарная
классификация)»

БГУИР 1-40 81 04

Магистрант:
гр. 858641
Кукареко А.В.

Проверил:
Стержанов М. В.

Минск, 2020

ХОД РАБОТЫ

Данные.

Набор данных DogsVsCats, который состоит из изображений различной размерности, содержащих фотографии собак и кошек. Обучающая выборка включает в себя 25 тыс. изображений (12,5 тыс. кошек: cat.0.jpg, ..., cat.12499.jpg и 12,5 тыс. собак: dog.0.jpg, ..., dog.12499.jpg), а контрольная выборка содержит 12,5 тыс. неразмеченных изображений. Скачать данные, а также проверить качество классификатора на тестовой выборке можно на сайте Kaggle -> <https://www.kaggle.com/c/dogs-vs-cats/data>.

Так как отправлять данные в <https://www.kaggle.com/c/dogs-vs-cats> уже нельзя, был найден альтернативный вариант: <https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition>

Задание.

1. Загрузите данные. Разделите исходный набор данных на обучающую, валидационную и контрольную выборки;
2. Реализуйте глубокую нейронную сеть с как минимум тремя сверточными слоями. Какое качество классификации получено?
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? Какой максимальный результат удалось получить на сайте Kaggle? Почему?

Результат выполнения:

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

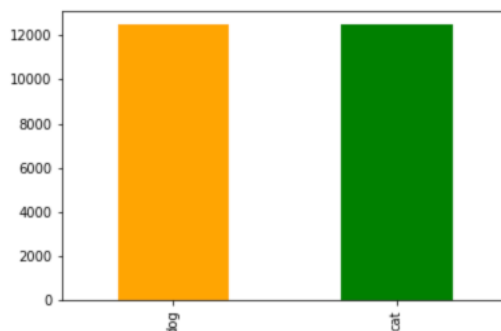


Рисунок 1 - кол-во изображений в каждом классе.

Как видно из рисунка 1 – данные в классах сбалансированы и мы можем применять метрику accuracy для определения точности классификатора.

Всего изображений 25 000. Они были разделены на:

- тренировочную выборку – 17 000 изображений;
- валидационную выборку – 4 000 изображений;
- тестовую выборку – 4 000 изображений.

Так же все изображения были приведены к размеру 180 x 180 пикселей, rgb (3 канала).

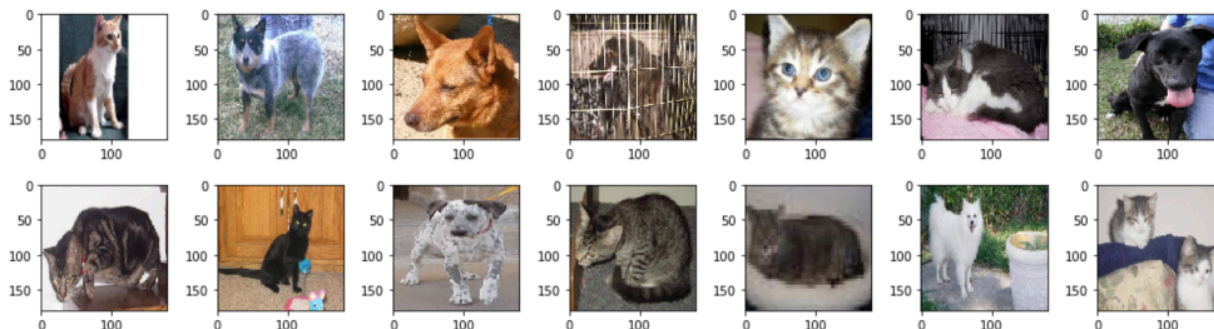


Рисунок 2 – пример данных из набора «dogs vs cats».

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 | - | 180x180x3 | - | - | - | - |
| 1 | Conv2d | 180x180 | 32 | 4 x 4 | 1 | ReLU |
| 2 | B. Norm. | - | - | - | - | - |
| 3 | Max Pool | 90x90 | 32 | 2 x 2 | 2 | - |
| 4 | Conv2d | 90x90 | 64 | 4 x 4 | 1 | ReLU |
| 5 | B. Norm. | - | - | - | - | - |
| 6 | Max Pool | 45x45 | 64 | 2 x 2 | 2 | - |
| 7 | Conv2d | 45x45 | 128 | 4 x 4 | 1 | ReLU |
| 8 | B. Norm. | - | - | - | - | - |
| 9 | Max Pool | 23x23 | 128 | 2 x 2 | 2 | - |
| 10 | Flatten | 67712 | - | - | - | - |
| 11 | FC | 128 | - | - | - | ReLU |
| 12 | Dropout | | | | | |
| 13 | FC | 128 | - | - | - | ReLU |
| 14 | Dropout | | | | | |
| Выходной | FC | 1 | - | - | - | Sigmoid |

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 20;
- batch size - 64;
- dropout - 0.3.

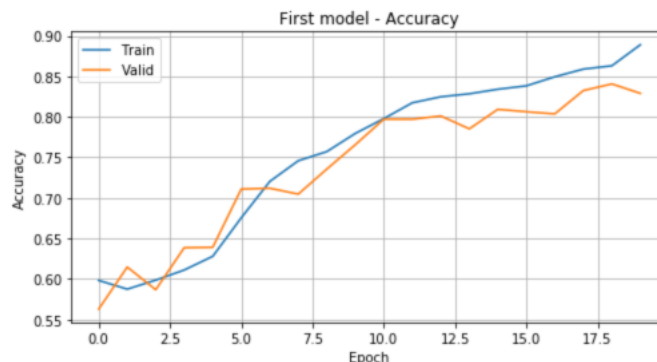


Рисунок 3 – график изменения ассигасы первой модели.

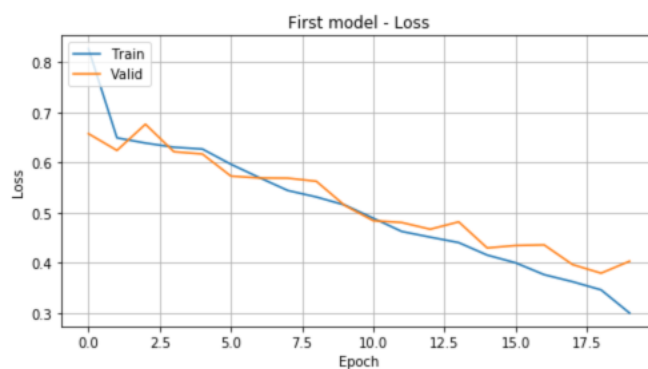


Рисунок 4 – график изменения loss первой модели.

На тестовой выборке модель показала следующий результат:

- loss - 0.4017;
- accuracy - 0.8245.

По графикам видно, что в районе 19 эпохи модель начала переобучаться, не смотря на dropout - 0.3.

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.

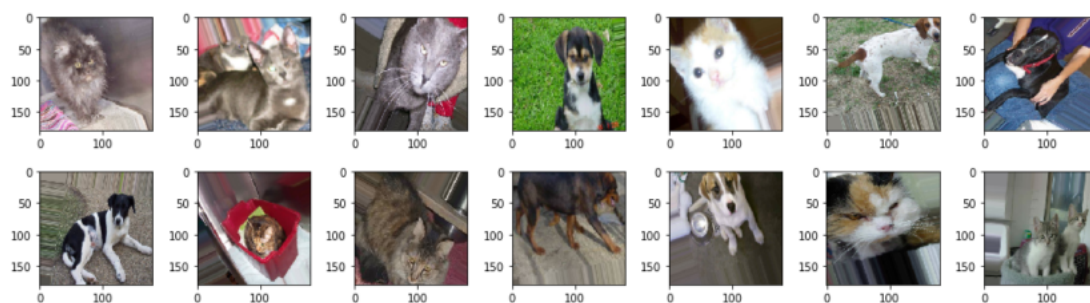


Рисунок 5 – пример аугментированных изображений в итерации 1.

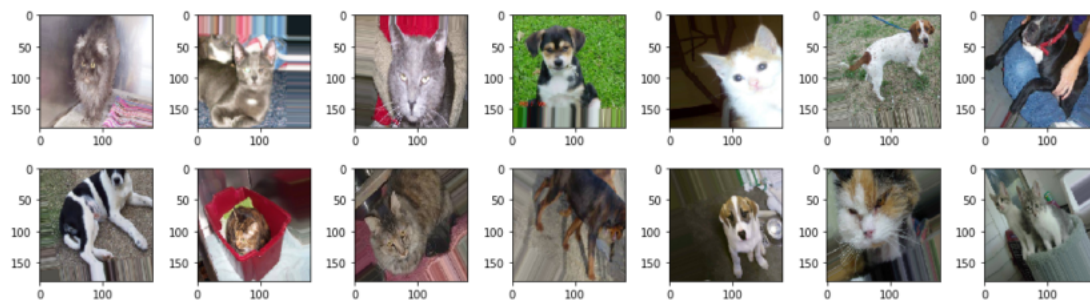


Рисунок 6 – пример аугментированных изображений в итерации 1.

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

Тренировка нейросети была запущена со следующими параметрами:

- epochs – 40;
- batch size - 64;
- dropout - 0.15.

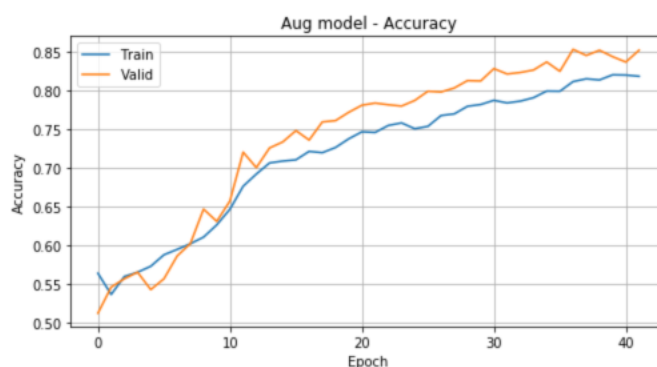


Рисунок 7 – график изменения ассигасу модели с применением аугментированных данных.

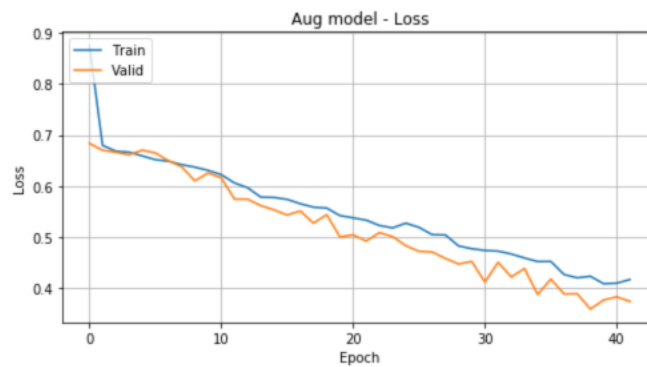


Рисунок 8 – график изменения loss первой модели с применением аугментированных данных.

На тестовой выборке модель показала следующий результат:

- loss - 0.3793;
- accuracy - 0.8472.

Для тренировки этой модели пришлось снизить показатель dropout с 0.3 до 0.15, так как модель не хотела обучаться на протяжении 10 эпох.

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 9 – архитектура сети vgg16.

У сети VGG16 были убраны последние 4 полносвязанных 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 | - | 180x180x3 | - |
| - | VGG15 | 5x5x512 | - |
| - | Flatten | 12800 | - |
| - | FC | 128 | ReLU |
| - | Dropout | | |
| - | FC | 128 | ReLU |
| - | Dropout | | |
| Выходной | FC | 1 | Sigmoid |

Так же было проведено 2 эксперимента:

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

Обучение моделей запускалось со следующими параметрами:

- epochs – 10;
- batch size - 64.

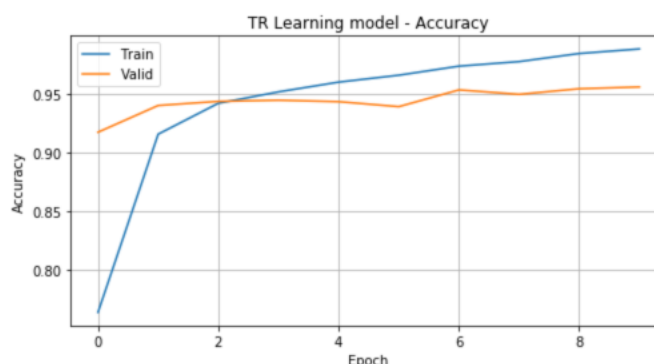


Рисунок 10 – график изменения ассигуры модели с применением передаточного обучения данных (без аугментации).

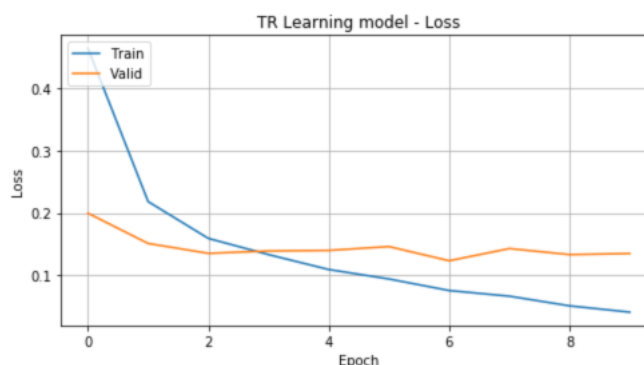


Рисунок 11 – график изменения loss первой модели с применением передаточного обучения данных (без аугментации).

На тестовой выборке модель показала следующий результат:

- loss - 0.1181;
- accuracy - 0.9575.

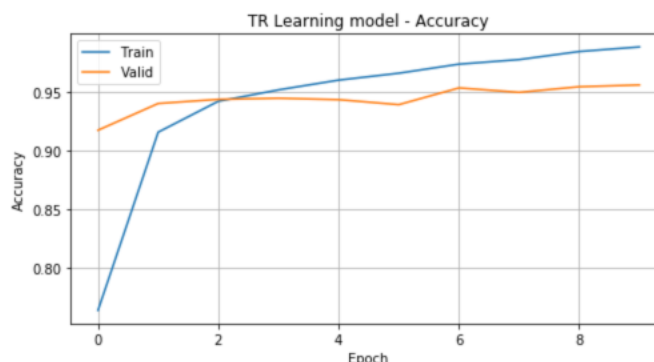


Рисунок 12 – график изменения ассигасу модели с применением передаточного обучения данных (с аугментацией).

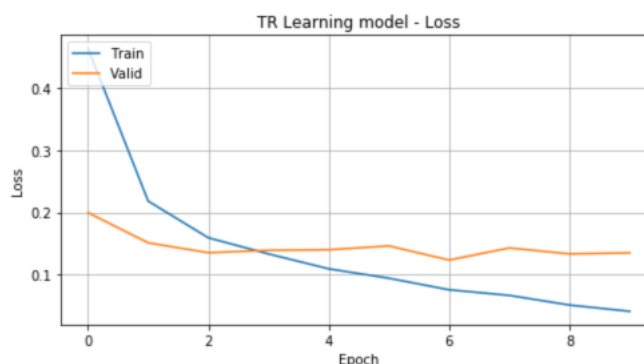


Рисунок 13 – график изменения loss первой модели с применением передаточного обучения данных (с аугментацией).

На тестовой выборке модель показала следующий результат:

- loss - 0.1188;
- accuracy - 0.9515.

После обучения всех моделей, были сформированы csv файлы и отправлены на Kaggle - <https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition> . Результаты kaggle можно посмотреть в таблице 3.

Таблица 3 – сравнение результатов на kaggle.

| Модель | kaggle score |
|--|--------------|
| Conv + original data | 0.41781 |
| Conv + augmented data | 0.36957 |
| Transfer learning vgg16 + original data | 0.44787 |
| Transfer learning vgg16 + augmented data | 0.51792 |

Вывод.

В ходе выполнения лабораторной работы я построил одну модель использующих сверточные слои для классификации изображений котов и собак. 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 аугментации данных позволяет снизить переобучение модели и повысить её точность, а техника «передаточного обучения» в некоторых случаях позволит значительно увеличить точность модели.