




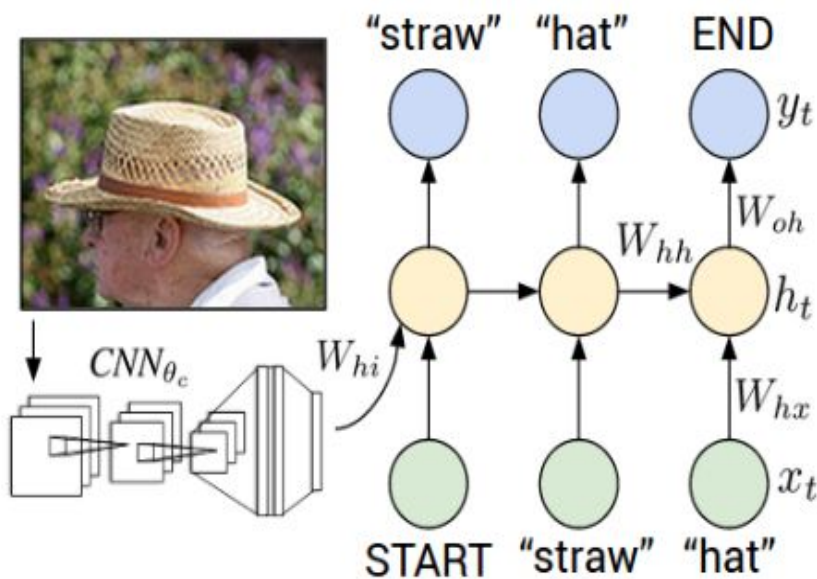
Image Captioning with PyTorch

By Denis Zankov

Содержание

- Постановка задачи
 - Описание датасета
 - Выбор модели
 - Инференс модели
 - Сферы применения
 - Дальнейшее развитие проекта
 - Выводы
- 

• Постановка задачи image captioning



- Image captioning – это модель encoder / decoder. На вход модели картинка, на выходе она возвращает текст с описанием того, что на ней изображено.

• Постановка задачи **image captioning**

- Encoder модель (CNN) получает на вход саму картинку и отдает вектор картинки. Вектор - это числовое описание картинки.
- Decoder модель (LSTM) берет этот вектор и генерировать текст. Вектор должен содержать всю необходимую информация для второй сети, чтобы она смогла нагенерить текста с описанием.

• Описание датасета

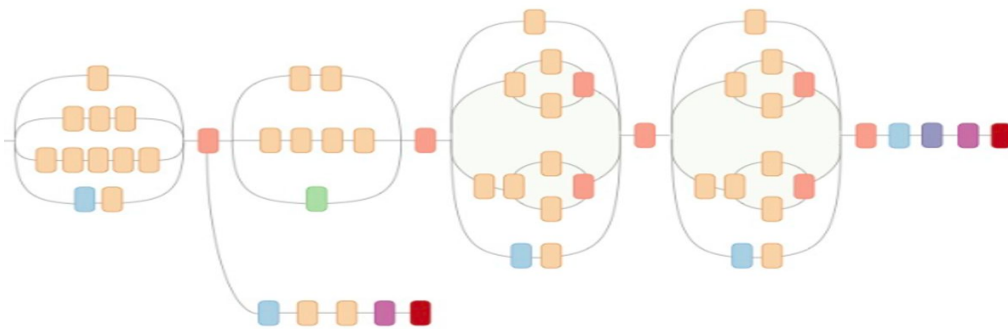
- <https://cocodataset.org/#captions-2015>
- Microsoft COCO Captions - это большой набор данных для обнаружения, сегментации, обнаружения ключевых точек и субтитров.
- Датасет состоит из 328К изображений.

• Выбор модели

- Для энкодера были протестированы предобученные архитектуры:
 - ResNet
 - AlexNet
 - VGG
 - Inception (Наилучший результат)
- Для декодера были обучены и протестированы:
 - GRU
 - LSTM
 - * Вывод делался не на основе метрик, а на основе адекватности предикта. Тестировал на собственных фотографиях.

. Выбор модели

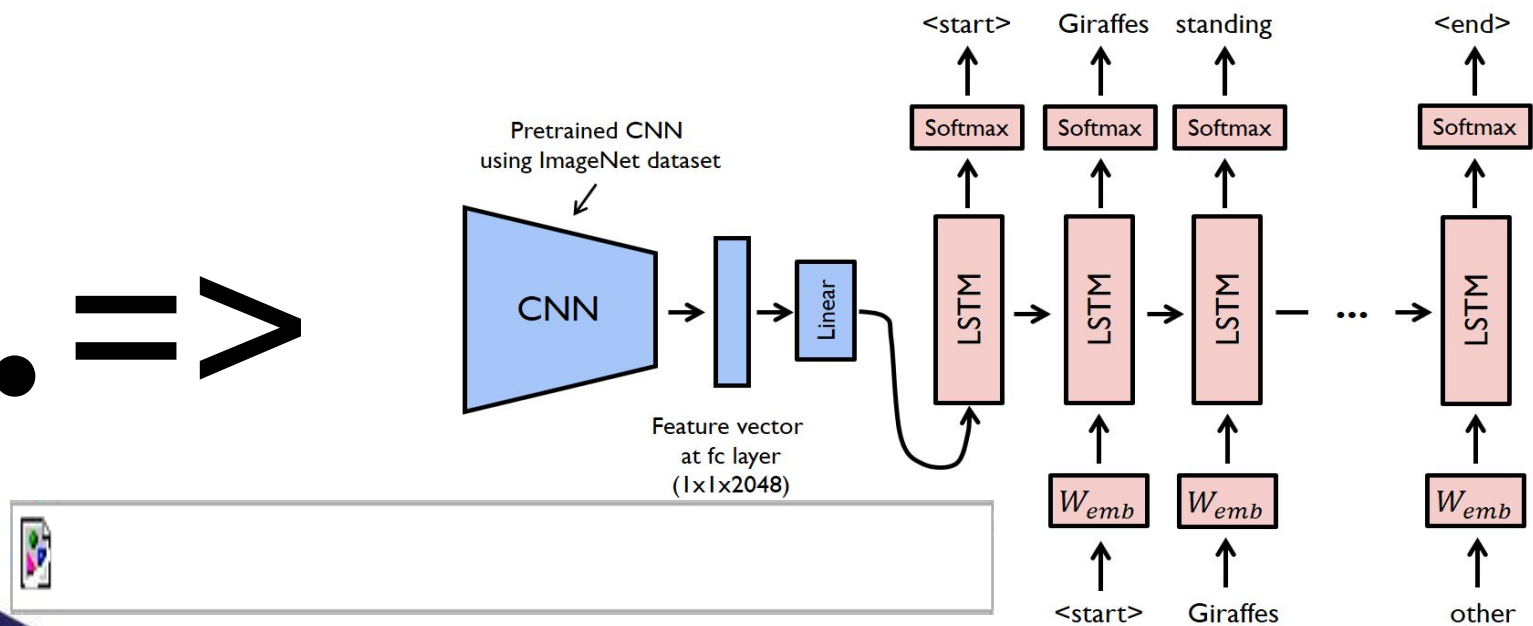
- . Beheaded Inception V3



• ==>

- . LSTM

• ==>



• Выбор модели – параметры обучения

- Network - sequence to sequence
- Criterion – CrossEntropyLoss
- Optimizer – Adam
- Some tricks – CLIP Gradient

```
: network = CaptionNet(cnn_feature_size=2048, hidden_size=256, vocab_size=vocab_size, num_layers=4).to(device)
: criterion = nn.CrossEntropyLoss(ignore_index=2)

: def compute_loss(network, image_vectors, captions_ix, device, criterion):
:     """
:     :param image_vectors: torch tensor с выходами inception. shape: [batch, cnn_feature_size]
:     :param captions_ix: torch tensor с описаниями (в виде матрицы). shape: [batch, word_i].
:
:     :returns: scalar crossentropy loss (neg log likelihood) for next captions_ix given previous ones
:     """
:
:     logits = network(image_vectors, captions_ix[:, :-1].type(torch.LongTensor).to(device))
:     loss = criterion(logits[0].view(-1, vocab_size), captions_ix.contiguous().view(-1).type(torch.LongTensor).to(device))
:
:     return loss

: optimizer = torch.optim.Adam(network.parameters()) # favourite one
```


• Выбор модели – генерация текста

- Чтобы генерировать различное описание на каждой итерации, жадный алгоритм не совсем подходит. Был написан простой алгоритм, в зависимости от номера слова от предложения, выбирает рандом из топ N токенов по вероятности, который показал большую эффективность:

```
import random
def get_top_n(n_iter):
    if n_iter < 4:
        return 4
    elif n_iter < 7:
        return 2
    else:
        return 1
```

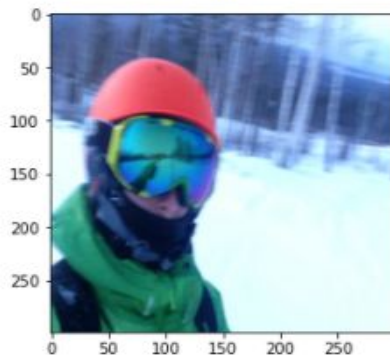
• Инференс модели (хорошо)

• Это я!

```
B [48]: img = plt.imread('img2.jpg')
img = np.array(Image.fromarray(img).resize((299,299))) / 255.

plt.imshow(img)
plt.show()

for i in range(10):
    print(' '.join(generate_caption(img, t=5.0)[1:-1]))
```



the person on the snowboard is going down the hill
there s no picture here to the side of the mountain
a person riding a snow board down a snow covered slope
there are a couple on the snow with a
an old woman standing in a snow covered area
there is two snowboarders that is skiing down the
an old photo with a person skiing down a snowy hill
a man on snow ski in the snow
an adult in a red jacket and a red
a skier in black jacket standing in the snow

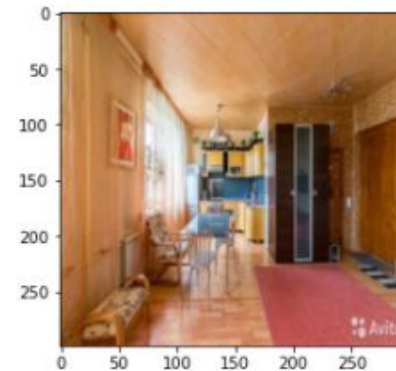
jetStarensShot

• Это моя квартира.

```
B [44]: img = plt.imread('img4.jpg')
img = np.array(Image.fromarray(img).resize((299,299))) / 255.

plt.imshow(img)
plt.show()

for i in range(10):
    print(' '.join(generate_caption(img, t=5.0)[1:-1]))
```



this kitchen is clean , and ready for us to use
an empty kitchen is decorated in a large room
a large room that is very large and clean
the view to the window of the living room
an image of two couches in the middle of a
this kitchen and house is empty and ready to use
an open couch and a table in a room
this is an apartment with an empty living room
this living area has a couch and a table
the living area shows the dining room and a living room

jetStarensShot

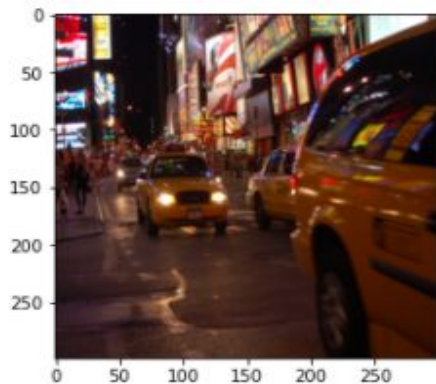
• Инференс модели (хорошо)

• Фото из путешествия

```
B [52]: img = plt.imread('img8.jpg')
img = np.array(Image.fromarray(img).resize((299,299))) / 255.

plt.imshow(img)
plt.show()

for i in range(10):
    print(' '.join(generate_caption(img, t=5.)[1:-1]))
```



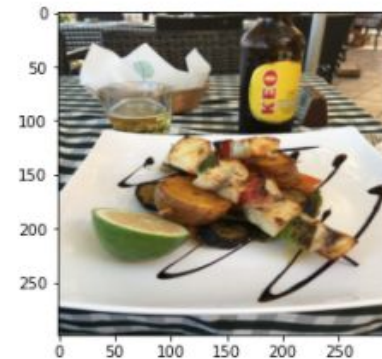
the car drives along a city street
the city bus has been changed for a parking meter
an image on street with a car parked on the
an empty city street with cars and cars
an image shot shows a car stop
a car is stopped at a stop light
two red cars are parked on the side of the street
the street signs is empty at night
two parking meters are on the side of the street
a parking meters on a city sidewalk with a car on the side

• Мой обед

```
img = plt.imread('img10.jpg')
img = np.array(Image.fromarray(img).resize((299,299))) / 255.

plt.imshow(img)
plt.show()

for i in range(10):
    print(' '.join(generate_caption(img, t=5.)[1:-1]))
```



two pieces with food on a white plate
a table that is filled for the meal
a plate filled with food and a cup of coffee
two pieces and food with some food on a plate
two pieces on a plate on a table with a sandwich
two sandwiches on the cob and a cup of coffee
there are a table with a plate of food on
a plate that is holding some food on it
a table filled that is covered with a sandwich and a cup of coffee
there 's an assortment if food on a plate

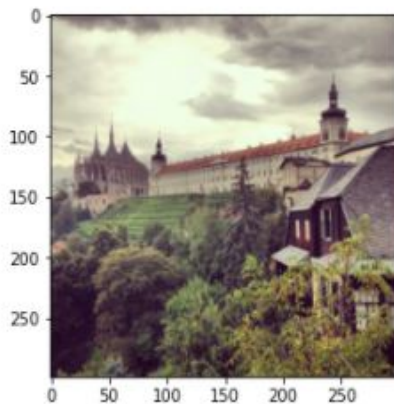
• Инференс модели (хорошо)

- Еще фото из путешествия

```
img = plt.imread('img12.jpg')
img = np.array(Image.fromarray(img).resize((299,299))) / 255.

plt.imshow(img)
plt.show()

for i in range(10):
    print(' '.join(generate_caption(img, t=5.)[1:-1]))
```



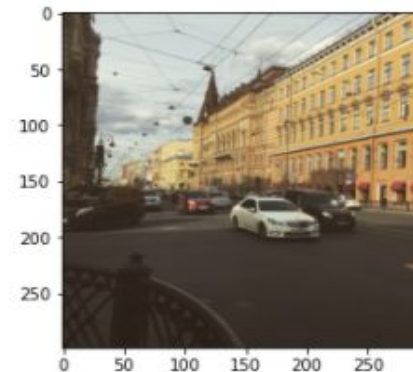
a large tower that is on top of a building
a clock on top a city road with a clock tower
a clock on an old brick tower with a clock on it
the clock tower in front of a building
there 's no image to a city street with a clock tower
the clock is standing in the middle of the city
an image shot with an old building with a clock tower
an old building is in front of a building
a tall brick tower in the middle of a city
an image shot shows the top end of a building with a clock tower

- Я в спб

```
img = plt.imread('img3.jpg')
img = np.array(Image.fromarray(img).resize((299,299))) / 255.

plt.imshow(img)
plt.show()

for i in range(10):
    print(' '.join(generate_caption(img, t=5.)[1:-1]))
```



cars driving past the street with a car on the side
a bus parked in a parking lot with a car on the side of
the street has a car and a car on the street
an empty street in a busy street with a car and a bus
the car rides through a street at night
a city with a lot of people on
an orange and yellow bus driving down a street
cars parked next to a car on a street
a bus is driving down the road with a car on the side
an orange truck is parked at the curb of a street

• Инференс модели (не очень)

• Пивной фестиваль

```
: img = plt.imread('img6.jpg')
img = np.array(Image.fromarray(img).resize((299,299))) / 255.

plt.imshow(img)
plt.show()

for i in range(10):
    print(' '.join(generate_caption(img, t=5.)[1:-1]))
```



a man with glasses is talking on a cell phone
the person in a red shirt has a cell phone
there 's an elephant that is sitting on the
there s no picture to look like a man
a young girl is holding up a cell phone
two young women sitting in front of a cell phone
there are three men that have a cell phone
the man in red is holding a cell phone
the young woman smiles as he talks on her cell phone
there is an older woman sitting in a chair with a cell

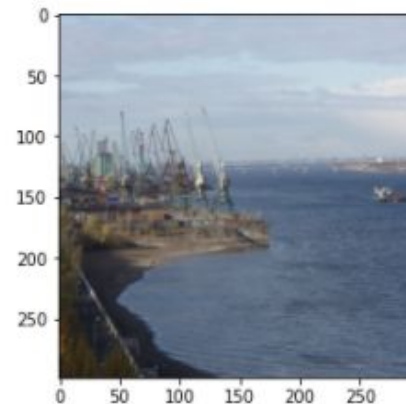


• Я в Перми

```
: img = plt.imread('img7.jpg')
img = np.array(Image.fromarray(img).resize((299,299))) / 255.

plt.imshow(img)
plt.show()


for i in range(10):
    print(' '.join(generate_caption(img, t=5.)[1:-1]))
```



several people standing in a body boat in the water
there are people that is standing in the sand
a bunch of water that are on a beach
several boats are docked in a large body of water
a bunch or water sitting on top of a beach
the water was on a beach of a beach
a group of boats sitting in a field
the view of the ocean and a body of water
the water was on a beach of a beach
the view from an ocean in a field with a boat



• Дальнейшее развитие проекта

- Использовать трансформеры в качестве декодера (а можно и енкодера).
 - Поднять вебсервис на AWS.
 - Потестить на видеопотоке.
 - Поработать над алгоритмом генерации текста.
 - Попробовать сделать обратную задачу.
 - На сгенерированный текст наложить генерацию звука.
- 

• **Выводы**

- Основная задача была получить адекватные описания к собственным фоточкам (которые сеть точно не видела) и это получилось!
 - Данное направление глубокого обучения можно использовать в социальных проектах (например, для слепых).
 - При разработке удалось поработать как со сверточными, так и с рекуррентными сетями.
- 