

Contents

1	Pretrained Network	2
2	Generative Adversarial Networks	2
2.1	CycleGAN	2
3	Image Captioning	2
4	Torch Hub	3

1 Pretrained Network

The pretrained network are referred to models/networks trained on ImageNet or subset of it. ImageNet is a dataset with about 14 million images which are maintained by Stanford. There was a competition called **ImageNet Large Scale Visual Recognition Challenge** (ILSVRC) where the training dataset consist of 1.2 million images. By using pretrained models, it means that we utilize models that their weights are trained on this specific dataset. These models can available in `torchvision.models`. The models module are named in a way that:

1. Uppercase names: Classes
2. Lowercase names: Instance of classes with its correspondence layers

If `pretrained` flag is set False, the weights will be random. If it True, ImageNet weights will be downloaded and initiated in the model.

Deep learning models are usually consists of sequential of filters and activation function following by fully connected layers.

Testing a trained model on a data is called **inference**. One thing to note is that the model should be set in `eval` model. This could cause different result because of layers such as Dropout and BatchNormalization.

2 Generative Adversarial Networks

Generative adversarial networks (GANs) consists of two parts:

1. Generator
2. Discriminator

Generators produce fake images from noise. The duty of discriminator is to label these made images as fake. The idea is to have a perfect generator which can trick discriminator (discriminator should not be able to detect fake images). In order to train this model, discriminator will inform the generator how good its images are.

2.1 CycleGAN

The usage is to change the domain of an image. For doing so, two generators and two discriminators are used. One main issue of GANs are instability of them. In order to meet this problem, they trained these two generators/discriminators at same time. The term of `Cycle` is chosen because of it.

3 Image Captioning

This type of models are consist of two parts:

1. Convolutional: image recognition
2. Recurrent: text generation

First part will detect objects which describe the image in a numerical way. The output of the first part is fed to the second part where **recurrent neural network** is used to generate sentences.

4 Torch Hub

From PyTorch1.0, **Torch Hub** is introduced which help in using other implemented models/networks than ones are defined in **torchvision**. For doing so, authors only needs to put a `hubconf.py` file in their repo. PyTorch will handle the other parts, and it will initiate that model with the trained weights. We should note that, Torch Hub does not install dependencies.