About Student

Name-Surname: Süleyman Gürsoy

Class: 3rd year

Student Number: 21527065

About Assignment

Class: BBM-418 Computer Vision Lab.

Subjects: Object Tracking with Regression Networks

Language: Python

1. What is Object Tracking?

The goal of object tracking is to keep track of an object in a video sequence. A tracking algorithm is initialized with a frame of a video sequence and a bounding box to indicate the location of the object we are interested in tracking. The tracking algorithm outputs a bounding box for all subsequent frames.

2. What is Neural Network?

A neural network is a type of machine learning which models itself after the human brain. This creates an artificial neural network that via an algorithm allows the computer to learn by incorporating new data.

While there are plenty of artificial intelligence algorithms these days, neural networks are able to perform what has been termed deep learning. While the basic unit of the brain is the neuron, the essential building block of an artificial neural network is a perceptron which accomplishes simple signal processing, and these are then connected into a large mesh network.

The computer with the neural network is taught to do a task by having it analyze training examples, which have been previously labeled in advance. A common example of a task for a neural network using deep learning is an object recognition task, where the neural network is presented with a large number of objects of a certain type, such as a cat, or a street sign, and the computer, by analyzing the recurring patterns in the presented images, learns to categorize new images.

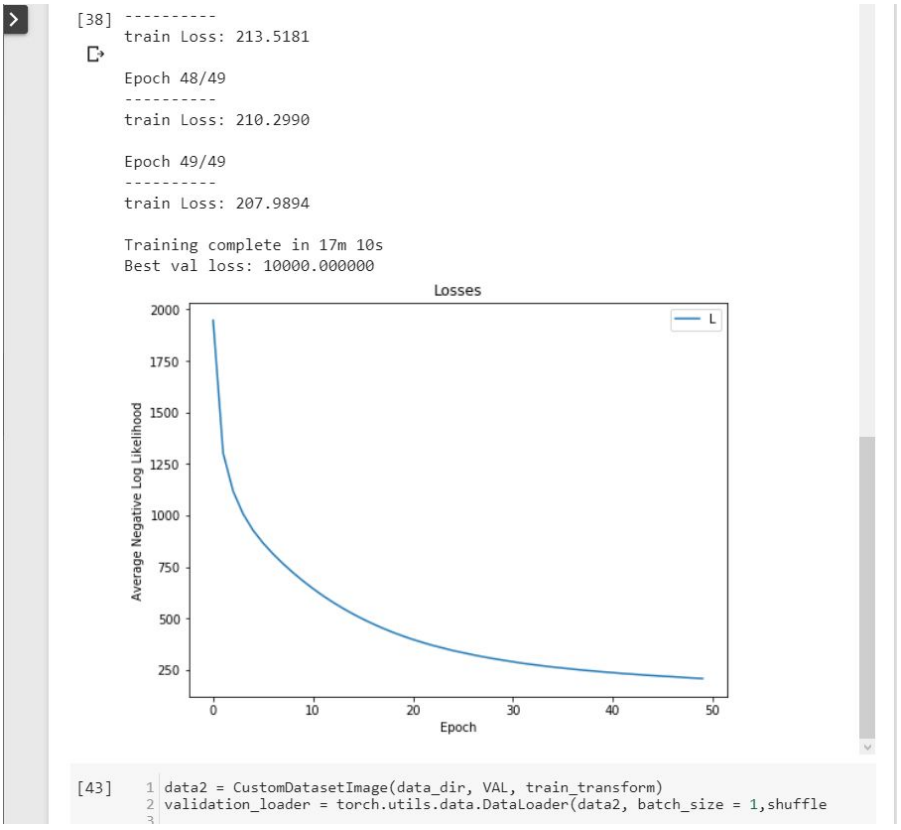
2.1 How Neural Networks Learn?

Unlike other algorithms, neural networks with their deep learning cannot be programmed directly for the task. Rather, they have the requirement, just like a child's developing brain, that they need to learn the information. The learning strategies go by three methods:

- Supervised learning: This learning strategy is the simplest, as there is a labeled dataset, which the computer goes through, and the algorithm gets modified until it can process the dataset to get the desired result.

- Unsupervised learning: This strategy gets used in cases where there is no labeled dataset available to learn from. The neural network analyzes the dataset, and then a cost function then tells the neural network how far off of target it was. The neural network then adjusts to increase accuracy of the algorithm.
- Reinforced learning: In this algorithm, the neural network is reinforced for positive results, and punished for a negative result, forcing the neural network to learn over time.

3. Experiments



```
  since = time.time()
  avg_loss = 0
  loss_test = 0
  total = 0
  vgg.to(device)
  for data in test_loader:
    vgg.train(False)
    vgg.eval()
    features= test_feature_extractor(new_vgg16, data)
    #print(len(features))
    for i in range(len(features)-1):
      feature1 = features[i][0]
      feature2 = features[i + 1][0]
      feature1.to(device)
      feature2.to(device)
      third = torch.cat((feature1,feature2),1)
      outputs = trained_model(third)
      loss =  criterion(outputs, features[i+1][1].float().to(device))
      loss_test += loss
      total += 1
      #print(total, loss_test)

  avg_loss = loss_test /total
  elapsed_time = time.time() - since
  print(0)
  print("Evaluation completed in {:.0f}m {:.0f}s".format(elapsed_time // 60, e
  print("Avg loss (test): {:.4f}".format(avg_loss))
```

```
[169]  eval_model(trained_model,criterion)
```

```
0
Evaluation completed in 0m 22s
Avg loss (test): 15656.2627
```

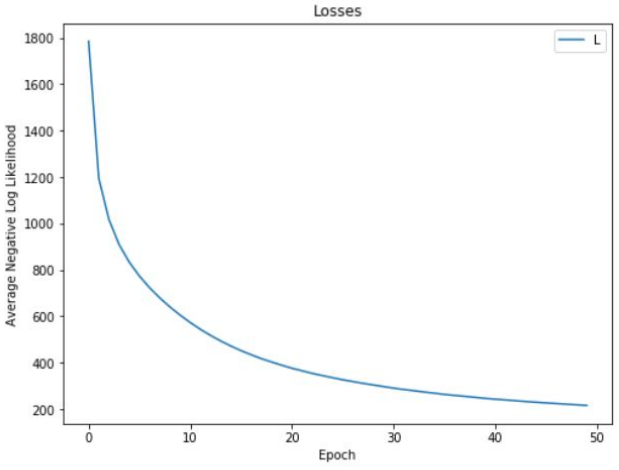
```
----------
[162] train Loss: 223.9093

Epoch 47/49
----------
train Loss: 221.1147

Epoch 48/49
----------
train Loss: 218.3937

Epoch 49/49
----------
train Loss: 215.7348

Training complete in 17m 23s
Best val loss: 10000.000000
```




```
[163]  data2 = CustomDatasetImage(data_dir, VAL, train_transform)
```