Pattern Recognition Theory 18794 Homework 3

Mengwen He (Alex)

December 1, 2016

Problem 1

Neural networks, or at least the very basic elements that made up these networks, *i.e.* the perceptron and ideas of learning machines, were proposed around the time the first computers were invented. Around the 80's and 90's neural networks, were not as popular. Why do you think so? And what has changed now that they have become so popular? Give a brief explanation or provide a discussion for each point. We are looking for two main reasons however, any valid reasons would be fine.

Answer:

1. The back-propagation (BP) algorithm did not work well on multiple hidden layers (except for certain architectures like CNN), often getting stuck in local optima. This led to poor results in benchmark datasets. The major issue identified was the initiation of weights in a deep network.

   This problem has been overcome recently by using layer-wise greedy unsupervised learning to initialize weights for each layer.

2. The computational power was not enough for training a deep neural network. For a little more complex problem, it could require more than one week to train a neural network around the 80's and 90's.

   However, with the advent of distributed and GPU systems, neural networks can be trained faster, and it may only cost one day to solve a very complex problem.

3. The well-labeled big dataset was not available around the 80's and 90's, but the neural network highly relays on the training data size.

   Now, with the improved technology, it becomes easier to collect, store, label, and analyze the big data from the Internet.

Problem 2

1. One of the key steps of a CNN convolution layer is weight sharing, *i.e.* the same kernel or filer is convolved over the entire image. Assume that the image is size w by h, and that the filter is m by n. Now once you convolve the filer with the image you get an output or response map. What is the size of the output or response map in terms of w, h, m, n?

   Answer:
   If the stride of the filter is 1, and the zero padding is 0, then the size of the output is$(w - m + 1) \times (h - n + 1)$.

2. Why is the weight sharing property useful? Mention and explain all the reasons you can think of. We are again looking for 2 main reasons, but feel free to suggest any valid reason. One hint for

the two reasons in order to give you a direction, would be parameters and spatial 2D information. Bonus: correctly explain the two reasons and come up with other reasons as well.

Answer:

(a) Sharing the weights and biases in each layer greatly reduces the number of parameters involved in a convolution network, and it relays on a reasonable assumption: if one feature is useful to compute at some spatial position, then it should also be useful to compute at a different position. The reduction of the number of parameters will result in faster training.

(b) Sharing the weights and biases enables the neural network to learn a single filter for a object no matter where the object appears in the image (translation invariance). Since translation invariance is fundamental to image-based recognition, sharing weights across image space is very useful in computer vision tasks.

Problem 3

Let's say you are trying to recognize a dog using a computer vision algorithm. You would want the algorithm to output the label "dog" no matter what kind the dog is, where the dog is in the image, and also what the dog is actually doing in the image. All these factors such as the kind of dog, the position of the dog in the image and the actual action of the dog can be said to be things you want to be invariant towards. They are in a bit more formal terms called nuisance transformations. Thus, in general, you would want your algorithm to be invariant to nuisance transformations in a pattern recognition task.

A CNN network has three main kinds of layers:

1. A convolution layer
2. A pooling layer
3. A fully connected

1. Out of the three layers which layer do you think explicitly enforces or helps with generating invariance to nuisance transformations? Please provide a brief explanation as to how and why do you think that particular layer can generate invariance?

   Answer:
   The convolution layer explicitly helps with generating invariance to nuisance transformations. Because the filter will do correlation operation across the entire input image, this will find the feature say of the dog no matter where the dog appears in the image.

2. In the midterm, we saw the virtual support method (VSV), which was trying to synthetically transform the support vector in order to produce different multiple transformed versions of the support vector points. The SVM is then run again on these support vectors. Thus the SVM is more invariant to the new transformations. Suggest one way to make a CNN more invariant to the nuisance transformations.

   Answer:
   Apply the nuisance transformations on the filter to generate virtual filters, and then use these virtual filters to extract the features in the convolution layer for the successive training.

3. In the following tasks, what nuisance transformations would you expect:

   (a) **Face recognition:** the position of the face in the image.

   (b) **Facial pose estimation:** people's different faces.

(c) **Speech recognition:** people's different voice tones.

(d) **Speaker identification:** the volume of the voice.

(e) **Hand written digit recognition:** people's different writing styles.

(f) **Handwriting recognition:** the stroke width.

Problem 4

Just to provide some brief intuition. The more complex the task, the more complex or deep the network you would want to use. Complexity of a task can be loosely said to be directly proportional to the amount of nuisance transformation that one might expect to exist in the data for that task.

1. In each of the following, you will be given two tasks or datasets. Please mention which one out of the two tasks/datasets would you employ a more complicated or a deeper CNN and why?

   (a) i. MNIST (`http://yann.lecun.com/exdb/mnist/`)
   ii. SVHN (`http://ufldl.stanford.edu/housenumbers/`)

   The SVHN needs a deeper CNN, because the data of SVHN has more rotational transformations than that of MNIST.

   (b) i. Object recognition
   ii. Face recognition

   Object recognition needs a deeper CNN, because the object data can be anything with variant transformations, but the face pose is always normally distributed.

   (c) i. Differentiating between two kinds of dogs
   ii. Differentiating between one kind of dog and one kind of tree

   Differentiating between one kind of dog and one kind of tree needs a deeper CNN, because in addition to the nuisance transformation of dogs, the second task will contain the nuisance transformation of trees.

2. One issue with deep learning and CNNs is that you need a lot of data to have the networks train enough to generalize well. Whereas, correlation filters and SVMs, and methods of the 90's and 2000's work well in cases where there is little data. It has been a challenge to apply deep learning to a few domains or tasks. Select the task in each case in which you think you'd use a SVM instead of a deep CNN, explain in brief why?

   (a) i. Kinds of cats
   ii. Classifying a particular kind of tumor using say a CT scan image

   It's better to use SVM for the tumor classifying, because the number of tumor CT scan images is relatively small. Very few people (compared to the entire population of human) will has tumor, and some of them will not go to hospital for variant reasons.

   (b) i. Age estimation of a person
   ii. Face recognition

   It's better to use SVM for the face recognition, because the number of one person's face images is limited. It's hard to collect a huge number of face images of one person in a short time.