Assignment 5: Convolutional Neural Networks

CS485/685 – Winter 2016

Out: March 24, 2016
Due: April 6, (11:59 pm), 2016. Submit an electronic copy of your assignment via LEARN. Late assignments may be submitted within 24 hrs for 50% credit.

1. **[50 pts]** For this question, you will experiment with convolutional neural networks, using the TensorFlow open source package. First, download and install TensorFlow from https://www.tensorflow.org/. Then go through the first three tutorials: 1) MNIST for ML beginners, 2) Deep MNIST for experts and 3) TensorFlow Mechanics 101. You will learn how to work with tensors (multi-dimensional arrays) and how to program classifiers based on softmax regression, shallow neural networks and convolutional neural networks to recognize digits in the MNIST dataset. Modify the code provided in the tutorials to answer the following questions. NB: You do not need to submit any code. Simply submit the results and your discussion of the results.

   (a) Compare the accuracy of softmax regression (MNIST for ML beginners), convolutional neural network (Deep MNIST for experts) and fully connected neural network with two hidden layers (TensorFlow Mechanics 101). Run the code from each tutorial as it is (except for the number of iterations that you can reduce to 1000 in order to avoid waiting too long). Report the training and testing accuracy for each algorithm and discuss the results (i.e., explain why some models perform better or worse than another models).

   (b) Compare the accuracy achieved by rectified linear units and sigmoid units in a convolutional neural network. Report the training and testing accuracy when running the convolutional neural network code (from the tutorial Deep MNIST for experts) as it is (i.e., with rectified linear units) and when replacing the rectified linear units by sigmoid units. You can reduce the number of iterations to 1000 to avoid waiting too long. Discuss the results (i.e., explain why some one type of unit performs better than the other).

   (c) Compare the accuracy achieved when varying the level of dropout in a convolutional neural network. Modify the convolutional neural network code (from the tutorial Deep MNIST for experts) to run with the $keep\_prob$ argument set to 0.25, 0.5, 0.75 and 1. You can reduce the number of iterations to 1000 to avoid waiting too long. Report the training and testing accuracy for each setting of $keep\_prob$. Discuss the results (i.e., explain how the parameter $keep\_prob$ affects the classifier and whether the results are as expected).

   (d) Compare the accuracy achieved when varying the number of hidden layers in fully connected neural networks. Modify the fully connected neural network code (from the tutorial TensorFlow Mechanics 101) to run with 1 hidden layer (150 nodes), 2 hidden layers (128 nodes in layer 1 and 32 nodes in layer 2) and 3 hidden layers (85 nodes in layer 1, 40 nodes in layer 2 and 25 nodes in layer 3). You can reduce the number of iterations to 1000 to avoid waiting too long. Report the training and testing accuracy for each number of layers. Discuss the results (i.e., explain how the the number of layers and the number of nodes per layer affects the results).

2. **[50 pts]** In object recognition, translating an image by a few pixels in some direction should not affect the category recognized. Suppose that we consider images with an object in the foreground on top of a uniform background. Suppose also that the objects of interest are always at least 10 pixels away from the borders of the image. Are the following neural networks invariant to translations of at most 10 pixels in some direction? Here the translation is applied only to the foreground object while keeping the background fixed. If your answer is

yes, show that the neural network will necessarily produce the same output for two images where the foreground object is translated by at most 10 pixels. If your answer is no, provide a counter example by describing a situation where the output of the neural network is different for two images where the foreground object is translated by at most 10 pixels.

(a) **[25 pts]** Neural network with one hidden layer consisting of convolutions (5x5 patches with a stride of 1 in each direction) and a softmax output layer.

(b) **[25 pts]** Neural network with two hidden layers consisting of convolutions (5x5 patches with a stride of 1 in each direction) followed by max pooling (4x4 patches with a stride of 4 in each direction) and a softmax output layer.