Introduction to Deep Learn, Fall 2017

Homework on Unit 3: 100 Total Points

Due: October 9th, 11:59PM

Submissions required electronically through Sakai. Non-code is required in pdf format, and code is required in a Jupyter notebook.

The purpose of this homework is to get a deeper understanding of the Convolutional Neural Network (CNN) architecture and practical intuition about its components, parameters, and their impact on performance.

Problem 1: Calculate receptive field given filter size, stride, pooling size (24 points)

The receptive field is defined as the region in the input space (part of an image) that an element of a CNN's feature map is representing. Note however that all elements (pixels) in the receptive field are not necessarily equally weighted, but relatively weighted by the elements of the convolutional filters and pooling layers. To get a better understanding of the influence of an image region to an element of a feature map please:

a) Derive an expression for the receptive field, in pixels, for one convolutional layer given its filter size and stride.
b) Extend the previous expression to the case where the convolution is followed by pooling operator given its size and stride.
c) Calculate the receptive field of VGG16 right before the first fully connected layer, for a 224x244x3 input image.

Recommended material: https://arxiv.org/pdf/1603.07285.pdf

Problem 2: Implementation of a 2-layer CNN with TensorFlow (24 points)

To gain experience with the implementation of CNNs in TensorFlow, please implement a 2-layer CNN followed by 2 fully connected layers as an architecture for a model to classify MNIST digits.

a) Specify the network without stride and 3x3 or 5x5 filter sizes. You can choose the number of filters in each layer and the number of hidden units of the first fully connected layer.
b) Calculate the receptive field of your 2-layer CNN for a 28x28 MNIST image.

Notes:

- For the implementation make sure of using TensorFlow's Core API. You are welcome to try other implementations using higher level APIs (*e.g.*, layers, Slim).
- You are only required to use the built-in cross entropy with logits and stochastic gradient descent. If you would like to try any of the optimization rules discussed earlier, you are welcome to.
- When initializing the weights of the model (not including biases) it is important to set their initial values to random values as discussed in class. You can use truncated normal distributions as discussed in class.

Problem 3: Adding pooling and dropout to a 2-layer CNN with TensorFlow (24 points)

To gain a better understanding of how pooling and dropout affect the performance of CNNs, please try the following:

a) Add 2x2 pooling layers after each convolutional layer for the specification in Problem 2.
b) Add dropout after the first fully connected layer for the specification in Problem 2. You can choose the probability of keeping (not setting to zero) hidden unit elements.

Notes:

- For the pooling layer, you are free to choose between max and average pooling, however we encourage you to use max pooling.

Problem 4: Performance comparison (24 points)

The goal is to compare the performance characteristics of different architectures at classifying MNIST digits. For this purpose, split the data into training and validation using the standard 60K/10K split. Please answer the following questions:

a) What is the validation accuracy of the CNN with and without pooling?
b) Did you observe any performance improvements after adding dropout?
c) How does the CNN model compare, in terms of performance, to the multi-class logistic regression and multi-class MLP from HW2?
d) How does the number of trainable parameters in the CNN models compare to that of the multi-class logistic regression and multi-class MLP from HW2?

Problem 5: Bookkeeping (4 points)

a) How many hours did this assignment take you? (There is No correct answer here, this is just an information gathering exercise)
b) Verify that you adhered to the Duke Community Standard in this assignment (https://studentaffairs.duke.edu/conduct/about-us/duke-community-standard), *i.e.*, write "I adhered to the Duke Community Standard in the completion of this assignment"