CS 224N: Assignment 2

RYAN MCMAHON TUESDAY 7TH FEBRUARY, 2017

Contents

Problem 1: Tensorflow Softmax (25 pts)

In this question, we will implement a linear classifier with loss function

$$J(\boldsymbol{W}) = CE(\boldsymbol{y}, softmax(\boldsymbol{x}\boldsymbol{W})) \tag{1.1}$$

Where $\boldsymbol{x}$ is a row vector of features and $\boldsymbol{W}$ is the weight matrix for the model. We will use TensorFlow's automatic differentiation capability to fit this model to provided data.

1.1 (a) Softmax in Tensorflow (5 pts)

Implement the softmax function using TensorFlow in `q1_softmax.py`. *Remember that*

$$softmax(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}} \tag{1.2}$$

Note that you may not use `tf.nn.softmax` *or related built-in functions. You can run basic (nonexhaustive tests) by running* `python q1_softmax.py`.

Answer:

See code: `~/code/q1_softmax.py`.

1.2 (b) Cross-Entropy Loss in Tensorflow (5 pts)

Implement the cross-entropy loss using TensorFlow in `q1_softmax.py`. *Remember that*

$$CE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{i=1}^{N_c} y_i log(\hat{y}_i) \tag{1.3}$$

where $\boldsymbol{y} \in \mathbb{R}^{N_c}$ is a one-hot label vector and N_c is the number of classes. This loss is summed over all examples (rows) of a minibatch. Note that you may ***not*** *use TensorFlows built-in cross-entropy functions for this question. You can run basic (non-exhaustive tests) by running* `python q1_softmax.py`.

Answer:

See code: `~/code/q1_softmax.py`.