odCS 229 Machine Learning, spring 2019

Homework 6:

Neural Networks

Due Saturday April 13, 11:59pm

Submit by the **blackboard system**

**Question1: (25 points) Property of derivatives of Error function (Exercise 5.6 and 5.7 of Bishop’s book)**

1. (5pts) Show the derivative of the error function

(5.21)

with respect to the activation ak for an output unit having a logistic sigmoid activation function

satisfies

(5.18).

1. (10 pts) Show the derivative of the error function

(5.24)

with respect to the activation ak for output units having a softmax activation function

satisfies

(5.18).

*Hint:* for each *,* its true label has  *,* and *.* That is to say, for the activation ak activated by an , the corresponding could be either 1 or 0.

1. (10 pts) How the derivative of the error function with respect to the activation helps on NN training? (updating the weights)

**Question2: (20 points) Vanishing gradient problem**

1. (10 points) What is the vanishing gradient problem? Can you demonstrate it and explain in mathematical ways?
2. (10 points) What are the most popular solutions to the vanishing gradient problem?

**Question3: (70 points) Implementation of NN (using Back-Propagation)**

**Code:**

Write your code in any programming language, and submit your results together with well-commented program source code. **An executable** file is **required**. You should **write your own code**, rather than just downloading existing code from the Internet.

**Data:**

Generate a set of data **points (x,y),** by choosing a **nonlinear function f(x)** and evaluating **y=f(x)+noise** for **random x** values, where each **x** is a **real vector of at least 2 elements** and each **y is a real scalar or vector**. As an alternative, you can usea data set you find online or are using in other research. You should clearly statewhat your data set is, or how you generated it.

**Note:**

For example, **x**=[x1; x2] is a 2-dim vector, f(**x**) can be x1+(x2)2+ (2\*x1-cos(x2)) 2.

f(**x**) should not be a simple one, like f(**x**)= f(x1) +f(x2), e.g., f(**x**)= x1 + (x2)2 +b, because, in such a case, the two inputs can be considered as one input is x1, and the other input is (x2)2. Then f(**x**) = f(**x** \* [1, 0]) + f(**x** \* [0, 1]). The obtained weights **W** will be a very simple set.

**Task:**

Implement a two-layer neural network with back-propagation.

The network should have **2 or more inputs**. The inputs connect to **M** neurons in the **hidden layer**, each of which takes a weighted sum of its inputs plus a bias and then applies the **hyperbolic tangent function (as the activation function)**.

The network should have 1 or more outputs. Each output of the network is a weighted sum plus bias of the outputs of the hidden layer (need no activation function or use **identity function f(x)=x** for the **output** because this is a regression problem).

During learning, both weights and biases change to decrease the mean squared error.

1) (5pts) **Describe all the parameters** you chose, including the number of inputs, outputs, and hidden neurons, the sizes of the initial random weights, learning rate etc.

2) (20 pts) Find a learning rate that allows it to learn to a small mean squared error. **Plot a figure of how the error decreases during learning.**

3) (20 pts) **Test** the NN you learned by a **different** set of data **points (x,y)** (different from training set, but y is still generated by f(x)+noise),what’s the error when comparing the **predicted** **yn** with the **true target** **y**? Give the mean and standard deviation of errors.

4) (25 pts) How will the training error and testing error be different if you re-train the NN by different initializations of weights? And how will they change if you set ***M*** (the number of hidden units) to be different values?

**Question4: (35 points)**

Implement **two types of gradient descent optimization strategies** discussed in class (**e.g., choosing two from momentum, AdaGrad, RMSProp, AdaDelta or Adam)**.

Run them and see if they learns faster, or learns a better solution (in terms of testing error). Describe whether they improved speed, accuracy, or both, and why you think that occurred. (It’s OK if you discover your “improvement” had no effect or even made it worse. The important thing is to test it and explain the results).