HOMEWORK 1: SHOW TEST RESULTS

1^{th} Edition, March 19, 2020

1 Show Homework

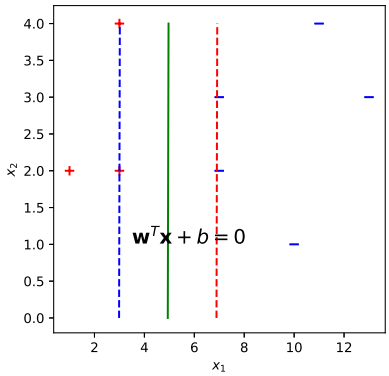
sadfasf
sdfdsf
sdf
Test citations:
[1-3] and [3].

1.1. Show Homework

Exercice 1: Problem 2.1

Test it!

Test inner subgraphs:



(a) $D = 1$

Here could be a graph.

(b) $D = 0.5$

Figure 1: Test graphs.

Theorem 1: example

check the theorem.

□

Proof:

Test subequations:

$$\frac{\partial \mathcal{L}(\mathbf{w},\ b)}{\partial \mathbf{w}} = \mathbf{w} + C\sum_i \frac{\partial \ell_i}{\partial \mathbf{w}}, \tag{1-1}$$

$$\frac{\partial \mathcal{L}(\mathbf{w},\ b)}{\partial b} = C\sum_i \frac{\partial \ell_i}{\partial b}, \tag{1-2}$$

Test codings:

```
# HyperPlate of SVM. It contains variables including w and b, and convert input x vector to a single value y(+-1).
with tf.name_scope('SVMPlate'): #Noted that the dimension of y must be 1, so the constants should be 1 dimensional.
    self.constrain = tf.constant(SVMPrimalSolution.Domain, dtype=tf.float32, shape=[1], name='Constrain')
    self.w = self.weight_variable([1, self.xDim], name='Weight')
    bias = self.bias_variable([1], name='Bias')
    self.subjection = tf.multiply(self.y, tf.matmul(self.w, self.x) + bias)
    tf.add_to_collection('Weight', self.w)
    tf.add_to_collection('Bias', bias)

@staticmethod
def weight_variable(shape, name=None):
    '''weight_variable generates a weight variable of a given shape.'''
    initial = tf.truncated_normal(shape, stddev=0.1)
    if name is not None:
        return tf.Variable(initial, name=name)
    else:
        return tf.Variable(initial)

@staticmethod
def bias_variable(shape, name=None):
    ''''bias_variable generates a bias variable of a given shape.'''
    initial = tf.constant(0.1, dtype=tf.float32, shape=shape)
    if name is not None:
        return tf.Variable(initial, name=name)
    else:
        return tf.Variable(initial)
```

□

2 References

[1] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2528-2535.

[2] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467-3478, Aug 2012.

[3] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295-307, Feb 2016.