50.021 -AI

Alex

Week 01: Discriminative ML - quick intro

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources.]

Homework - we start in class - momentum, RMSProp, AdaDelta

1. Implement the momentum training rule as described in lecture in the file `quadform.py`; edit the method `update_m`. You can set the value of the `self.momentum` factor in the `__init__` method of `quadform2D`. At the bottom of the file, you will find a list of testing functions that illustrate the effects on performance.

2. Implement the RMSProp training rule as described in lecture in the file `quadform2.py`; edit the method `update_r`. You can set the value of the `self.rmspropconst` factor in the `__init__` method of `quadform2D`. Note that we have provided an initialization for the first step. At the bottom of the file, you will find a list of testing functions that illustrate the effects on performance.

3. Implement the AdaDelta training rule as described in lecture in the file `quadform3.py`; edit the method `update_a`. You can set the value of the `self.rmspropconst` factor in the `__init__` method of `quadform2D`. At the bottom of the file, you will find a list of testing functions that illustrate the effects on performance.

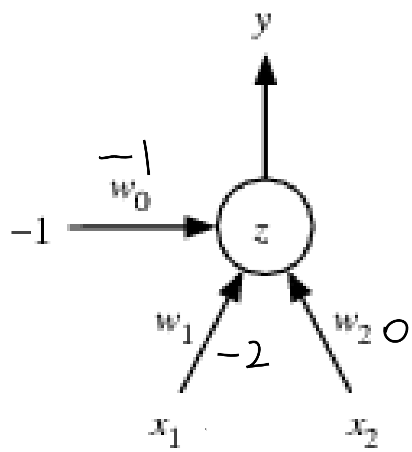
Homework - another activation I

For this problem, we will consider the simple type of unit shown below. The output of the unit $g(z)$ is computed as follows:

$$g(z) = \begin{cases} z & \text{if } |z| < 1 \\ sign(z) & \text{otherwise} \end{cases}$$

$$z = -w_0 + w_1 x_1 + w_2 x_2$$

$W_0 = -1, \quad W_1 = -2, \quad W_2 = 0$



X_1	X_2	Z	Y
0	0	1	Positive
0	1	1	Positive
1	0	-1	Negative
1	1	-1	Negative

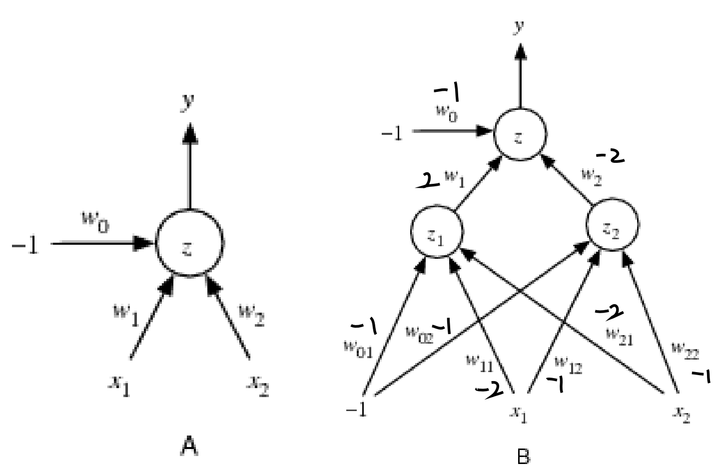
We can use this type of unit to classify our inputs by assigning any input for which the output is greater than or equal to 0 as positive and for which the output is less than 0 to negative.

Given the four data points: Positive: $(0,0)$, $(0,1)$ Negative: $(1,0)$, $(1,1)$, choose weights for this unit so the weights w_0, w_1, w_2 that can separate these points.

Homework - another activation II

Given the four data points: Positive: $(0,0)$, $(1,1)$ Negative: $(0,1)$, $(1,0)$ (not the same as above!) and the following two networks, using the type of unit defined above.

X_1	X_2	Z_1	Z_2	Z_3	Y
0	0	1	1	1	Positive
1	1	0	0	1	Positive
1	0	0	1	0	Negative
0	1	0	1	0	Negative



- Task: Can either of the two networks above successfully classify them? Explain why (and which ones) or why not.
- Task: Derive the stochastic gradient-descent rule for a single unit of this type. This learning rule is incremental, meaning it updates

the weights for each individual training point. Thus, we will consider the error at a single training example, i. Assume that the input vector has been augmented so that the zero-th diomension of the vector x: x_0 is always -1 for every datapoint. That way we don't need a separate rule for the constant weight terms. By that we obtain: $z = w_0x_0 + w_1x_1 + w_2x_2$.

Assume we use the usual quadratic error. Note that the derivative of $g(z)$ is discontinuous, which is undesirable, but answer the question ignoring this issue.