Homework 7 solutions

1. On the class website is some data that was taken from 4 measurements of 150 samples of three kinds of iris. If you download `IrisData.mat` from the class website, and in Matlab type `load IrisData`, two matrices will appear: X will be 150×4 are represent the four measurements per flower (so it has size "number of points" $\times$ "dimension"). The desired output is in the array Y. If the flower is a "class 1" iris, the corresponding Y has row $(1, 0, 0)$. Class 2 is $(0, 1, 0)$, and Class 3 is $(0, 0, 1)$.

   (Question to think about: Why are the "targets" not the integers $1, 2, 3$?)

   "SOLUTION": Using these targets would impose a metric on the classes- By that I mean that it would imply that Class 3 is closer to Class 2 (one "unit" away) than Class 1 (2 "units" away). Better to use targets that would make the distances between classes the same (unless there is a reason to do otherwise).

   There is a sample script that was started for you online. The only thing that is missing is the part where we send the data to Widrow-Hoff to get W, b.

   It would be hard to assess the classification using a graph, and so we compute a "confusion matrix". Read the code over and see if you can figure out what the confusion matrix is.

   - Try training with the data in the order given, with a learning rate of about $0.5-0.1$, and about 500 iterations. Record what you see in the "confusion matrix".

     SOLUTION: Probably get something like

     $$\text{Confuse} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

   - Try again, but re-order the data randomly (note: a Matlab command that might be useful is `randperm`). Again record the confusion matrix.

     ```
     %Random Selection (be sure to change domain AND range!)
     idx=randperm(150);
     [W,b,err]=wid_hoff1(X(idx,:),Y(idx,:),0.1,500);
     ```

     $$\text{Confuse} = \begin{bmatrix} 1.00 & 0 & 0 \\ 0 & 0.64 & 0.36 \\ 0 & 0.10 & 0.90 \end{bmatrix}$$

   - Try to understand what you see-

     SOLUTION: In keeping the data "in order", we were training only class 1 too long- in detriment to trying to classify class 2 especially. In randomizing the order of the data, we are giving equal relevance to the data in all the classes (in terms of what numerical values will be assigned to the weights and biases).

2. • Write a Matlab function `myfunc` that will input a vector $\mathbf{x} \in \mathbb{R}^3$ and output two things- the scalar $f(\mathbf{x})$ and the vector $\nabla f(\mathbf{x})$ for

$$f(\mathbf{x}) = 5x_1^2 - x_1x_2 + 6x_2^2$$

• Use your previous Matlab function to illustrate gradient descent. Write a script file that has you starting at the point $(1,1)$

SOLUTION: You need turn in only the script file. I'll include it here for completeness:

```
%Starting point:
x=[1;1];

%Number of iterations and "learning rate":
iters=10;
lr=0.1;

for j=1:iters
    [Y(j),dy]=myfunc(x);
    x=x-lr*dy;
end
%***************
 function [f,df]=myfunc(x)

 f=5*x(1)^2-x(1)*x(2)+6*x(2)^2;
 df=[10*x(1)-x(2); -x(1)+12*x(2)];
```

3. Write the solution to exercise 8 from the appendix.

Let $f(x,y) = 3x^2 + xy - y^2 + 3x - 5y$. Then the gradient and Hessian are below:

$$\nabla f = [6x + y + 3, \quad x - 2y - 5] \qquad Hf = \begin{bmatrix} 6 & 1 \\ 1 & -2 \end{bmatrix}$$

(a) f increases most quickly from $(1,1)$ in the direction of $[10,-6]^T$
f decreases most quickly from $(1,1)$ in the direction of $-[10,-6]^T$
The rate of change of f in that direction is $\sqrt{10^2 + (-6)^2} = \sqrt{136}$

(b) The Hessian is computed above.

(c)

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \begin{bmatrix} 6 & 1 \\ 1 & -2 \end{bmatrix} \mathbf{x} + [3, \quad -5]\mathbf{x}$$

(d) • The critical point (stationary point) is where $A\mathbf{x} + \mathbf{b} = \mathbf{0}$, or

$$\mathbf{x} = -A^{-1}\mathbf{b} \approx \begin{bmatrix} -0.077 \\ -2.539 \end{bmatrix}$$

or $x_1 = -1/13$, $x_2 = -33/13$.

• The eigenvalues of A are -2.12 and 6.12, so there is a saddle point at this critical point.