Midterm

Due: Tuesday, October 30, 12:30pm.

Your solutions to these problems should be uploaded to ELMS as a single pdf file by the deadline. As with problem sets, you may turn in the solution up to two days late, with a penalty of 10% per day, and you should only upload one version of your solutions.

This exam is individual and open book. You may consult any reference work. If you make specific use of a reference outside those on the course web page in solving a problem, include a citation to that reference. So, for example, if you consult a general reference to understand the meaning of VC-dimension, you don't need to cite that. But if you find a reference that contains a specific result that you draw on in solving the problem on VC-dimension, please cite that. You may discuss the course material in general with other students, but you should work on the solutions to the problems on your own. So, again, you can discuss the definition of VC-dimension with other students. You can even go over the VC-dimension of a different problem with other students. But you should not work on the specific problem in this midterm with other students.

It is hard to write questions in which every possibility is taken into account; as a result, there may be sometimes "trick" answers that are simple and avoid addressing the intended problem. Such trick answers will not receive credit. As an example, suppose we said, use the chain rule to compute $\frac{\partial z}{\partial x}$ with $z = \frac{7}{y}$ and $y = x^2$. A trick answer would be to say that the partial deriviative is not well defined because y might equal 0. A correct answer might note this, but would then give the correct partial derivative when $y \neq 0$.

1. Suppose we have a neural network that has a loop in it (see Figure 1). So when it is given an input, it follows a circular computation. We can index these computations by time. So we can say that the input is a d-dimensional vector, x. This is constant over time. At $t = 1$ we compute the first unit:

$$\begin{aligned} z_1(1) &= w^1 \cdot x + b^1 \\ a_1(1) &= \max(0, z_1(1)) \end{aligned}$$

where w^1 is a vector of length d, and b^1 is a scalar bias. Then we have:

$$\begin{aligned} z_2(1) &= w^2 a_1(1) + b^2 \\ a_2(1) &= \max(0, z_2(1)) \end{aligned}$$

Here w^2 is just a scalar weight. Similarly we have:

$$\begin{aligned} z_3(1) &= w^3 a_2(1) + b^3 \\ a_3(1) &= \max(0, z_3(1)) \end{aligned}$$

However, the ouput of the second unit feeds back into the first unit. So, at subsequent times, for $t > 1$ we have:

$$\begin{aligned} z_1(t) &= w^1 \cdot x + b^1 + w^4 a_2(t-1) \\ a_1(t) &= \max(0, z_1(t)) \end{aligned}$$

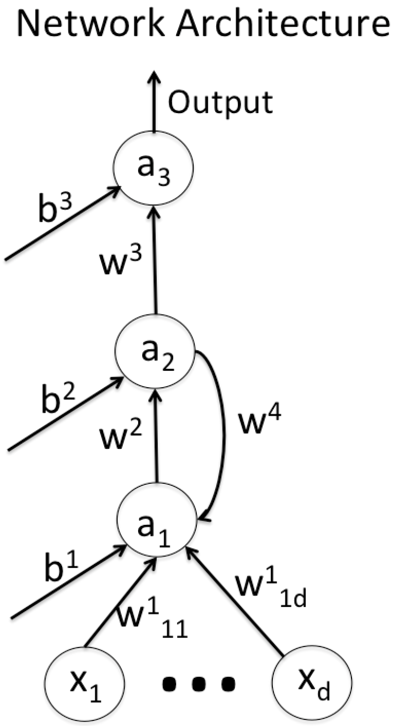

Figure 1: A network with a loop, used in Problem 1

with the next steps largely unchanged:

$$\begin{aligned} z_2(t) &= w^2 a_1(t) + b^2 \\ a_2(t) &= \max(0, z_2(t)) \\ z_3(t) &= w^3 a_2(t) + b^3 \\ a_3(t) &= \max(0, z_3(t)) \end{aligned}$$

We repeat these computations T times. Then the output of the network is $a_3(T)$. Suppose we train this network with a loss of:

$$L(x, y) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - a_3(x^i, T))^2$$

This is just the standard regression loss. y_i denotes the i'th label, x^i denotes a vector containing the i'th input, and $a_3(x^i, T)$ denotes the network output after T iterations, with an input of x^i.

- If we want to train the network using gradient descent, what is the gradient if we have $T = 2$?

- Suppose we use the network by using $T = \infty$, and running the network until it converges. What is the gradient of the loss in this case?

2. Convolution.

   - True or false: suppose we convolve an image twice with any pair of 3×3 filters. Then there exists a 5×5 filter such that convolution with this filter is equivalent to convolution with the two 3×3 filters. Either show that this is true or give an example of two 3×3 filters that cannot be represented by a 5×5 filter.
   - True or false: suppose we convolve an image once with a 5×5 filter. Then there exist two 3×3 filters such that convolution with these two filters is equivalent to convolution with the 5×5 filter. Either show that this is true or give an example of a 5×5 filter that cannot be represented by two 3×3 filters.
   - Let G_σ be a 1D Gaussian filter with s standard deviation of σ. Let $u(t) = (G_\sigma * \cos)(t)$, that is, the cosine function filtered with the Gaussian. If $u(0) = .9$, what is the value of $u(\pi/8), u(\pi/4), u(\pi/2)$?

3. Suppose we have a shallow neural network with one hidden layer. This network has two layers of weights, W^1 and W^2, along with bias terms. Suppose we randomly initialize the weights from a Gaussian distribution. However, during training we fix the weights in layer 2 (W^2). Only the weights in the first layer, and all the bias terms are allowed to vary. Is this network able to approximate any function with arbitrary accuracy, as the number of hidden units grow? Prove that your answer is correct. Note that part of this problem is to precisely formulate what you are going to prove in a reasonable way.

4. Consider a set of 2D points with labels $+1$ or -1. A hypothesis, $h \in \mathcal{H}$, assigns all points in the 2D plane to one of the two classes.

   - Suppose $\mathcal{H}$ is the set of all linear separators. That is, for any w, b there is a hypothesis h, such that $h(x) = sign(wx + b)$. What is the set of five 2D planar points that has the largest Rademacher complexity? What is the complexity of these points? What is the set of five planar points with the lowest Rademacher complexity? What is the complexity of those points?
   - Suppose the hypotheses in $\mathcal{H}$ are based on the intersection of two half-planes. That is, for any choice of w_1, w_2, b_1, b_2 there exists a hypothesis h such that $h(x) = 1$ if and only if $w_1x + b_1 > 0$ and $w_2x + b_2 > 0$. What is the VC-dimension of $\mathcal{H}$?

   For each of these problems prove, or at least provide a convincing argument as to why your answer is correct.

5. Using your own words, convince us that you understand how batch normalization works. Your explanation should be much less than one page long.

6. **Challenge Problem (extra credit):** We saw the following theorem in class:

   Suppose the function $f : \mathcal{R}^n \to \mathcal{R}$ is convex and differentiable, and that its gradient is Lipschitz continuous with constant $L > 0$, i.e. we have that $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$ for any x, y. Then if we run gradient descent for k iterations with a fixed step size $t \leq 1/L$ it will yield a solution $f^{(k)}$ which satisfies:

   $$f(x^{(k)} - f(x^*) \leq \frac{\|x^{(0)} - x^*\|_2^2}{2tk}$$

   Consider a neural network with an input layer of dimension d, one hidden layer containing k hidden units and a single output unit. The network has a RELU operation in the hidden layer. We train the network with the standard regression loss:

   $$L(x, y) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

   using n training examples. Suppose further that all of the weights and biases in the network, and all of the input values and labels are constrained to have values between -1 and 1.

   Let θ stand for a vector containing all the network parameters. Consider $R(\theta)$ to be a function that maps the network parameters to the loss that the network would produce on all the training data.

   - Is the gradient of R Lipschitz continuous with constant $L > 0$? Either prove that it is not, or prove that it is and give the lowest value of L that you can show this is true for.