ECE-GY 6143: Introduction to Machine Learning
Final Exam, Spring 2019

Prof. Sundeep Rangan

Answer ALL questions. Exam is closed book. No electronic aids. But, you are permitted two cheat sheets, two sides each sheet. Any content on the cheat sheet is permitted. Part marks are given. If you do not remember a particular python command or its syntax, use pseudo-code and state what syntax you are assuming.

You may answer on the exam sheet or in the provided blue book. But, if you use a blue book, please indicate on the exam paper. You must turn in the exam paper at the end of the exam. You may keep your cheat sheet.

Best of luck!

1. *SVM.* Consider an SVM classifier,

$$\widehat{y} = \begin{cases} 1 & \text{if } z > 0, \\ -1 & \text{if } z \geq 0, \end{cases} \qquad z = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b,$$

for the kernel function,

$$K(\mathbf{x}, \mathbf{x}_i) := \begin{cases} 1 & \text{if } \|\mathbf{x} - \mathbf{x}_i\|^2 \leq r^2, \\ 0 & \text{if } \|\mathbf{x} - \mathbf{x}_i\|^2 > r^2, \end{cases}$$

where $r > 0$ is some parameter and $\|\mathbf{w}\|^2$ denotes the squared norm $\|\mathbf{w}\|^2 = w_1^2 + w_2^2$. You are given four training samples.

i	1	2	3	4
x_{i1}	0	1	0	2
x_{i2}	0	0	2	2
y_i	1	-1	-1	1

(a) Find parameters r, α_i and b such that the SVM classifier makes no errors on the training data. Use at most two non-zero values of α_i.

(b) Given the choice of parameters in (a), draw the region of (x_1, x_2) where the classifier predicts $\widehat{y} = 1$.

(c) Complete the following python function `predict` that computes a vector of outputs `yhat` for a data matrix `X`. You must provide the other arguments needed. For full credit, avoid for loops.

```
def predict(X,...):
    ...
    return yhat
```

2. *Neural Networks.* Consider a neural network, with $N_i = 2$ input units, $\mathbf{x} = (x_1, x_2)$, $N_h = 3$ hidden units and one output unit for binary classification:

$$z_j^{\mathrm{H}} = \sum_{k=1}^{N_i} W_{jk}^{\mathrm{H}} x_k + b_j^{\mathrm{H}}, \quad u_j^{\mathrm{H}} = \begin{cases} 1 & \text{if } z_j^{\mathrm{H}} > 0 \\ 0 & \text{if } z_j^{\mathrm{H}} \leq 0, \end{cases} \quad j = 1, \ldots, N_h$$

$$z^{\mathrm{O}} = \sum_{k=1}^{N_h} W_k^{\mathrm{O}} u_k^{\mathrm{H}} + b^{\mathrm{O}}, \quad \widehat{y} = \begin{cases} 1 & \text{if } z^{\mathrm{O}} > 0 \\ 0 & \text{if } z^{\mathrm{O}} \leq 0. \end{cases}$$

(a) Suppose that

$$\mathbf{W}^{\mathrm{H}} = \begin{bmatrix} 0 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{b}^{\mathrm{H}} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

For each hidden unit $j = 1, 2, 3$, draw the regions of inputs (x_1, x_2) where $u_j^{\mathrm{H}} = 1$.

(b) Find a vector of output weights $\mathbf{W}^{\mathrm{O}}$ and bias b^{O} such that:

- $\mathbf{x} = (0, 0)$, $(2, 0)$, $(1, 2)$ are classified as $\widehat{y} = 0$; and
- $\mathbf{x} = (1, 0.5)$ is classified as $\widehat{y} = 1$.

3. *Backpropagation.* Consider the model with D-dimensional inputs $\mathbf{x} = (x_1, \ldots, x_D)$ and M-dimensional outputs $\mathbf{y} = (y_1, \ldots, y_M)$, given by

$$\widehat{y}_m = \sum_{\ell=1}^{L} B_{\ell m} z_\ell, \quad z_\ell = \sum_{j=1}^{D} x_j A_{j\ell}, \quad m = 1, \ldots, M,$$

with parameters $A_{j\ell}$ and $B_{m\ell}$.

(a) You are given training samples $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})$ and $\mathbf{y}_i = (y_{i1}, \ldots, y_{iM})$. Write $\widehat{y}_{im}$ in terms of the inputs x_{ij}.

(b) Draw the computation graph between the data, parameters and loss function using the loss function,

$$J = \sum_{i=1}^{N} \sum_{m=1}^{M} (y_{im} - \widehat{y}_{im})^2.$$

(c) Show how to compute $\partial J / \partial A_{j\ell}$ from $\partial J / \partial z_{i\ell}$.

4. *CNN kernels.* A systems has five cameras that each take 256×192 dimension gray scale images. A mini-batch of training samples consists of 100 samples, with each sample having the five camera images.

   (a) What is the shape of a tensor X representing the mini-batch of training data.

   (b) A first layer of a CNN performs a convolution with 20 output channels and 3×3 kernels W. Write the equation relating the input X and output Z. What is the shape of W and output Z. Assume the convolution is performed on the valid pixels.

   (c) Describe a possible kernel W that detects a difference between camera 2 and camera 3, but ignores cameras 0,1, and 4. The difference output should be in output channel 0.

5. *CNN sub-sampling.* Consider a 1D convolution following by sub-sampling,

$$z[j] = \sum_k w[k]x[j+k], \quad u[m] = z[sm],$$

for some stride parameter $s > 0$. That is, u takes every s samples of z.

(a) If x is 250 milliseconds of audio sampled at 20 kHz, how many samples are in x?

(b) If x has length 1000, w has length 10 and $s = 4$, how many output samples are in u assuming the convolution is only computed on the valid samples.

(c) Given a gradient $\partial J/\partial u[m]$, how do you compute $\partial J/\partial w[k]$?

6. *PCA.* You are given python arrays with PCA data:

- `mu`: A `100`-dim vector representing the mean of the data; and
- `V`: A `(100,5)` array with the 5 top PCs of the data.
- `lam`: A `5`-dim vector of eigenvalues.

(a) Given a vector `z` of PC coefficients, write a few lines of python code to reconstruct the data `x`.

(b) Now, suppose you are given only the first 80 of the 100 coefficients of a vector `x`. Write a few lines of python code to estimate the PCA coefficients `z`. You can assume you have a function,

```
w = lstsq(A,b)  # Solves the least-squares solution to b=A.dot(w)
```

(c) Continuing in part (b), how would you estimate the remaining 20 coefficients of `x`?

7. *K-means.* You are given five data samples:

i	1	2	3	4	5
x_{i1}	0	1	0	2	2
x_{i2}	0	0	1	2	3

(a) Draw the five points.

(b) Starting with $K = 2$ cluster centers at $(0, 0)$ and $(1, 0)$, what are the cluster assignments and new cluster centers after one iteration of K-means?

(c) Suppose you want to use clustering for outlier detection. You find cluster means $\boldsymbol{\mu}_i$, $i = 1, \ldots, K$ on the training data. Then, given a new data $\mathbf{x}$ and a threshold t, you declare $\mathbf{x}$ an outlier if $\|\mathbf{x} - \boldsymbol{\mu}_i\| \geq t$ for all i. Complete the following function to implement the outlier detection on a matrix of data `X`. The output is `out[i]=1` if the sample `X[i,:]` is an outlier, and `out[i]=0` otherwise. You must specify the other inputs of your function. Avoid for loops for full credit.

```
def outlier_detect(X, ...):
    ...
    return out
```