**ENSE 885AU Deep Learning**

**Assignment A02**

**Multiclass Logistic Regression**

(Due date: Friday October 8th 2021, 23:59)

*Instructions:*

Math component

1. On separate pieces of paper or using a software with math input, answer the questions for this assignment.
2. Scan or convert your drawings to Adobe PDF format.
3. Login to URCourses ([https://urcourses.uregina.ca](https://urcourses.uregina.ca)) and submit your answers to Assignment A02.

Coding component

4. Login to Snoopy (**snoopy.engg.uregina.ca** / **142.3.105.92**).
5. Create your program using any text editor of your choice (e.g. **vi / emacs**)
6. Name your files using the following convention:

   Main program file *"A"+number+username+"Q"+number+".py"* (e.g. **A02jon123Q3.py** if your username is jon123 and you are solving question 3)

7. Run your program and test that it works correctly (e.g. **python A02jon123Q3.py**)
8. Submit all your python source code files (Type **~ense885au/bin/submit A02 A02jon123Q3.py A02jon123Q4.py**)

---

1. (Math) Suppose for multiclass logistic regression

$$p(x|y=i) = N(x|\mu_i, \Sigma),$$

Compute

$$a_i = \ln[p(x|y=i)p(y=i)].$$

Is it still linear? What if $p(x|y=i)$ has different covariance? That is,

$$p(x|y=i) = N(x|\mu_i, \Sigma_i),$$

2. (Math) Recall that the SVM objective can be simplified as minimizing

$$\tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j$$

subject to $\alpha_i \geq 0 (\forall i)$, and $\sum_{i=1}^{n} \alpha_i y_i = 0$. Suppose the optimal solution is $\alpha^* = ({\alpha_1}^*, {\alpha_2}^*, \cdots, {\alpha_n}^*)$. Show that the margin $\gamma$ satisfies

$$\frac{1}{\gamma^2} = \sum_{i=1}^{n} {\alpha_i}^*$$

3. (Coding) Download the benchmark dataset MNIST from http://yann.lecun.com/exdb/mnist/. Implement multiclass logistic regression and try it on MNIST.

   *Comments: MNIST is a standard dataset for machine learning and also deep learning. It's good to try it on one layer neural networks (i.e., logistic regression) before multilayer neural networks. Downloading the dataset from other places in preprocessed format is allowed, but practicing how to read the dataset prepares you for other new datasets you may be interested in. Also, it is recommended to try different initializations and learning rates to get a sense about how to tune the hyperparameters (remember to create and use validation dataset!).*

4. (Coding) Consider the $l_2$-regularized logistic regression. That is, add to the logistic regression loss a regularization term that represents $l_2$ norm of the parameters. More precisely, the regularization term is

$$\lambda \sum_i \left( \|w^i\|^2 + \|b^i\|^2 \right)$$

   where $\{w^i,\ b^i\}$ are all the parameters in the logistic regression, and $\lambda \in \mathbb{R}$ is the weight for the regularization. Typically, $\lambda$ is about $C/n$ where $n$ is the number of data points and $C$ is some constant in [0.01; 100] (need to tune $C$). Run the regularized logistic regression on MNIST, and compare the results to those in Problem 3.

**The End**