ECE C147/C247, Winter 2022 **Homework #2**
Neural Networks & Deep Learning Prof. J.C. Kao
University of California, Los Angeles; Department of ECE TAs: Y. Li, P. Lu, T. Monsoor, T. Wang
M. Kleinman, C. Zheng

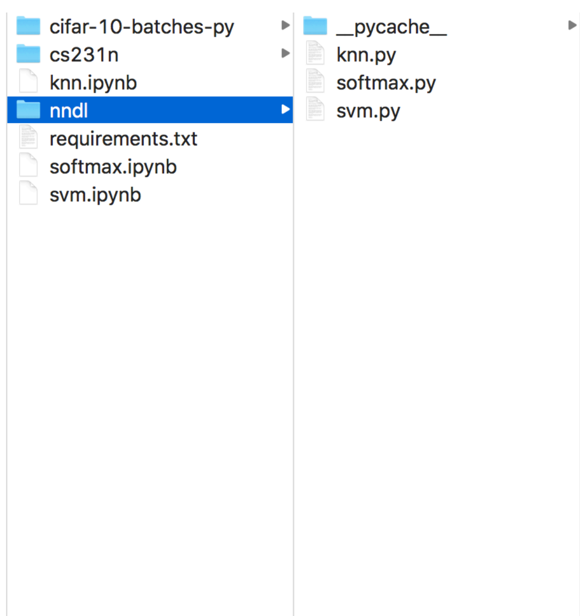
Due Monday, 24 Jan 2022, by 11:59pm to Gradescope.
100 points total.

1. (20 points) ***k*-nearest neighbors.** Complete the k-nearest neighbors Jupyter notebook. The goal of this workbook is to give you experience with the CIFAR-10 dataset, training and evaluating a simple classifier, and k-fold cross validation. In the Jupyter notebook, we'll be using the CIFAR-10 dataset. Acquire this dataset by running:

```
wget http://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
tar -xzvf cifar-10-python.tar.gz
rm cifar-10-python.tar.gz
```

If you don't have `wget` you can simply go to: `https://www.cs.toronto.edu/~kriz/cifar.html` and download it manually.

We have attached a screenshot of the paths the files ought to be in, in case helpful (though it should be apparent from the Jupyter notebook).



Print out the entire workbook and related code sections in knn.py, then submit them as a pdf to gradescope.

2. (40 points) **Softmax classifier gradient.** For softmax classifier, derive the gradient of the log likelihood.

   Concretely, assume a classification problem with c classes

   - Samples are $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})$, where $\mathbf{x}^{(j)} \in \mathbb{R}^n$, $y^{(j)} \in \{1, \ldots, c\}$, $j = 1, \ldots, m$
   - Parameters are $\theta = \{\mathbf{w}_i, b_i\}_{i=1,\ldots,c}$
   - Probablistic model is

     $$\Pr\left(y^{(j)} = i \mid \mathbf{x}^{(j)}, \theta\right) = \text{softmax}_i(\mathbf{x}^{(j)})$$

     where

     $$\text{softmax}_i(\mathbf{x}) = \frac{e^{\mathbf{w}_i^T \mathbf{x} + b_i}}{\sum_{k=1}^{c} e^{\mathbf{w}_k^T \mathbf{x} + b_k}}$$

   Derive the log-likelihood $\mathcal{L}$, and its gradient w.r.t. the parameters, $\nabla_{\mathbf{w}_i}\mathcal{L}$ and $\nabla_{b_i}\mathcal{L}$, for $i = 1, \ldots, c$.

   Note: We can group $\mathbf{w}_i$ and b_i into a single vector by augmenting the data vectors with an additional dimension of constant 1. Let $\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$, $\tilde{\mathbf{w}}_i = \begin{bmatrix} \mathbf{w}_i \\ b_i \end{bmatrix}$, then $a_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + b_i = \tilde{\mathbf{w}}_i^T \tilde{\mathbf{x}}$. This unifies $\nabla_{\mathbf{w}_i}\mathcal{L}$ and $\nabla_{b_i}\mathcal{L}$ into $\nabla_{\tilde{\mathbf{w}}_i}\mathcal{L}$.

3. (40 points) **Softmax classifier.** Complete the Softmax Jupyter notebook. Print out the entire workbook and related code sections in softmax.py, then submit them as a pdf to gradescope.