For $t = 0$, please fix $w^{(t)}$ to have the correct dimensionality. The strange indexing $x_{(t\%N)+1}$ simply says to iterate through the data in the order provided above. Note the second column of each row should equal the last column of the previous row.

3. (**35 points**) First, let $f(x)$ be a K-dimensional feature vector, i.e., $f(x) \in \mathbb{R}^K$. The particular meaning of x doesn't matter for this question: it could be an image, or a text document, or collection of robotic sensor readings. In class, we discussed one method of turning a (binary) linear model into a probabilistic (binary) classifier: given vector weights $w \in \mathbb{R}^K$, we pass our linear scoring model $w^\intercal f(x)$ through the sigmoid function $\sigma(z) = \frac{1}{1+\exp(-z)}$ and use the following decision function:

$$\begin{cases} \text{output "class 1"} & \text{if } \sigma(w^\intercal f(x)) \geq 0.5 \\ \text{output "class 2"} & \text{if } \sigma(w^\intercal f(x)) < 0.5. \end{cases} \quad \text{[Eq-1]}$$

This question considers the multi-class extension of that model.

Second, assume we have L classes, with corresponding weights θ_l ($1 \leq l \leq L$) per class. We can group these weights into a matrix $\theta \in \mathbb{R}^{LxK}$. Notice that each θ_l is a K-dimensional vector.

Consider a probabilistic linear classification model $p(\hat{y}|x)$, computed as

$$p(y = \hat{y}|x) = \frac{\exp(\theta_{\hat{y}}^T f(x))}{\sum_{y'} \exp(\theta_{y'}^T f(x))}, \quad \text{[Eq-2]}$$

where $\hat{y}$ and y' are potential values for the label.

(a) Argue that a binary instantiation (i.e., $L = 2$) of model [Eq-2] provides the same decision function as [Eq-1]. In arguing this, you do not need to prove it in a strict mathematical sense, but you need to make it clear that you understand it. (Imagine how you would explain it to a classmate.)

We showed in class that optimizing the MAP estimate is the "right" thing to do when we want to minimize the empirical posterior 0-1 classification loss. Remember that $\log$ is monotonic: if $f(a) < f(b)$, then $\log f(a) < \log f(b)$. As such, this is also called **maximizing the conditional log-likelihood**:

$$\arg\max_y p(y|x_i) = \overbrace{\arg\max_y}^{\text{maximizing}} \underbrace{\log p(y|x_i)}_{\text{the conditional log-likelihood}} \quad \text{[Eq-3]}$$

Given the correct label $y_i^\star$, the above equation will be maximized for $y = y_i^\star$, i.e., $\log p(y_i^\star|x)$ will be the maximum value (across possible y arguments). Using a one-hot representation of the label,[1] we can define the **cross-entropy loss**, which is the dot product between $\vec{y^\star}$ and the *vector* of log conditional probability values $\log p(y|x_i)$:

$$\ell^{\text{xent}} = -\sum_{j \in \mathcal{Y}} \vec{y^\star}[j] \log p(y = j|x_i). \quad \text{[Eq-4]}$$

[1] It is sometimes beneficial to interpret a correct label $y^\star$ in a *one-hot format* $\vec{y^\star}$. This one-hot vector $\vec{y^\star}$ is a vector the size of $\mathcal{Y}$: each coordinate $\vec{y^\star}[j]$ corresponds to one of the j items of $\mathcal{Y}$. In a one-hot format, all entries are 0 except for the coordinate corresponding to $y^\star$. As an example, consider the case of rolling a 1, 2, 3, 4, 5, or 6 (represented by y) from a weighted six-sided die (where this die outcome represents our label), given some input x: to represent a correct roll of $y^\star = 4$, a reasonable one-hot equivalent is $\vec{y^\star} = (0, 0, 0, 1, 0, 0)$.

(b) Show that, for the probabilistic classifier [Eq-2], maximizing the conditional log-likelihood ([Eq-3]) is the same as minimizing cross-entropy loss ([Eq-4]).[2]

(c) Given N data points $\{(x_1, y_1), \ldots, (x_N, y_N)\}$, formulate the empirical risk objective for [Eq-2] using cross-entropy loss. You can think of this as filling out the following template,

$$\underset{\clubsuit}{\text{opt}} \underbrace{\frac{1}{N} \sum_{i=1}^{N} \diamondsuit}_{\mathcal{L}(\clubsuit)}, \qquad \text{[Eq-5]}$$

where you must specify

- opt: the optimization goal (e.g., max, min, or something else)
- $\clubsuit$: the variables or values being optimized
- $\diamondsuit$: the function (and its arguments) being optimized.

(d) Derive the gradient $\nabla_{\clubsuit} \mathcal{L}(\clubsuit)$.

Classification: Implementation, Experimentation, and Discussion

The next three questions lead you through building, experimenting with, reporting on the performance of, and analyzing a multiclass perceptron classifier. The **core deliverables** for these questions are:

1. any implementations, scripts, and serialized model files needed to compile and run your code; and
2. a written report discussing
   - your implementations, including any design decisions or difficulties you experienced [from questions 4-5];
   - evidence and analysis of internal development/experimentation on the perceptron model on the *development* set [from question 5]; and
   - (optionally, for extra credit) results and analysis of a full comparison on the *test* set [from 6].

The data we'll be using is the MNIST digit dataset: in total, there are 70,000 "images" of handwritten digits (each between 0-9). The task is to predict the digit $y \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ from a 28x28 input grayscale image x. 60,000 of these are allocated for training and 10,000 are allocated for testing. Do **not** use the 10,000 testing portion until question 6.

Get the data from `https://www.csee.umbc.edu/courses/graduate/678/spring23/materials/mnist_rowmajor.jsonl.gz`. This is a gzipped file, where each line is a JSON object. Each line has three keys: "image" (a row-major representation of each image), "label" (an int, 0-9), and "split" (train or test). Each image x is represented as a 784 ($= 28 \times 28$) one-dimensional array (row-major refers to how each image x should be interpreted in memory).

[2] You may be wondering what are the predicted items in cross-entropy loss. The cross-entropy loss measures the correctness of probabilities corresponding to particular classes, against hard, "gold standard" (correct) judgments. Therefore, the predicted items are actually probabilities, from which you can easily get a single discrete prediction.