Problem 1 (15 Points): Logistic Regression

We have a domain with two features, X_1 and X_2, and a class variable Y. Y is binary with values $\langle \text{false}, \text{true} \rangle$. We would like to train a logistic regression model whose weights are w_0, w_1, and w_2.

$$P(Y = \text{false} \mid X_1, X_2) = \frac{1}{1 + e^{w_0 + w_1 \cdot X_1 + w_2 \cdot X_2}}.$$

Assume the current weights are as follows: $w_0 = 0.5, w_1 = -1, w_2 = 1$. You will calculate the partial gradient of conditional log-likelihood (CLL) with respect to each of the weights: w_0, w_1, and w_2.

Below is the training dataset. For your convenience, the predicted probabilities of each class have already been calculated. Complete the last three columns of the table by calculating the partial gradient of CLL with respect to each weight for each data point. Show your detailed answer.

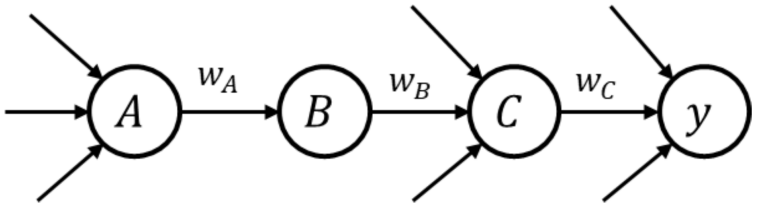
$$\text{CLL} = \log P(Y \mid X_1, X_2)$$

$$\frac{\partial \text{CLL}}{\partial w_i} = (Y - P(\text{true} \mid X_1, X_2)) \cdot \frac{\partial z}{\partial w_i}, \quad z = w_0 + w_1 \cdot X_1 + w_2 \cdot X_2$$

Data ID	X_1	X_2	Y	$P(\text{false} \mid X_1, X_2)$	$P(\text{true} \mid X_1, X_2)$	$\frac{\partial \text{CLL}}{\partial w_0}$	$\frac{\partial \text{CLL}}{\partial w_1}$	$\frac{\partial \text{CLL}}{\partial w_2}$
d_1	-4	-2	true	0.08	0.92			
d_2	-2	-1	false	0.18	0.82			
d_3	0	0	false	0.38	0.62			
d_4	$+3$	$+2$	true	0.62	0.38			
d_5	$+1$	-1	false	0.82	0.18			

Problem 2 (10 Points): Backpropagation

Here is a part of a larger feedforward neural network. The output is the variable y. This is a classification task where y is binary and has a sigmoid activation function.



The activation at A is `tanh`, at B is `sigmoid`, at C is `tanh`, and at y is `sigmoid`. The weight from one node to another is given in the diagram. We want to minimize the cross-entropy loss:

$$E = -(1 - t)\ln(1 - y) - t\ln(y),$$

where t is the true target value.

Assume the current weights are: $w_A = 2, \quad w_B = -3, \quad w_C = 4$.

Assume the current activations for a given data point are: $A = -0.7, \quad B = 0.6, \quad C = 0.23, \quad y = 0.8$.

Assume the true target value for this data point is $t = 0$. (Note: These activations are based on everything that goes into these nodes and the rest of the network, which is not shown.)

Show your detailed answers for the below questions:

1. **2 points.** What is the partial gradient of the error with respect to w_C, i.e., $\frac{\partial E}{\partial w_C} = ?$
2. **3 points.** What is the partial gradient of the error with respect to w_B, i.e., $\frac{\partial E}{\partial w_B} = ?$
3. **5 points.** What is the partial gradient of the error with respect to w_A, i.e., $\frac{\partial E}{\partial w_A} = ?$

Hints

- The derivative of the sigmoid function $\sigma(x)$ is:

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x)).$$

- The derivative of the hyperbolic tangent function $\tanh(x)$ is:

$$\tanh'(x) = 1 - \tanh(x)^2.$$

Use these derivatives in your calculations to propagate errors back through the network.

Problem 3 (15 Points): Mixture Models: Mixtures of Bernoulli Distributions

Consider a set of D binary variables x_i, where $i = 1, \cdots, D$, each of which is governed by a Bernoulli distribution with parameter μ_i, so that

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i}(1 - \mu_i)^{(1-x_i)},$$

where $\mathbf{x} = [x_1; x_2; \cdots, x_D]$ and $\boldsymbol{\mu} = [\mu_1, \mu_2; \cdots, \mu_D]$.

Now let us consider a finite mixture of these distributions given by

$$p(\mathbf{x}|M, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\boldsymbol{\mu}_k),$$

where $M = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$, $\boldsymbol{\pi} = \{\pi_1, \pi_2, \cdots, \pi_K\}$, and

$$p(\mathbf{x}|\boldsymbol{\mu}_k) = \prod_{i=1}^{D} \mu_{ki}^{x_{ki}}(1 - \mu_{ki})^{(1-x_{ki})}.$$

Show that:

1. **6 Points.** The mean of this mixture distribution is given by

$$\mathbb{E}[\mathbf{x}] = \sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k.$$

2. **9 Points.** The covariance of this mixture distribution is given by

$$\text{cov}[\mathbf{x}] = \sum_{k=1}^{K} \pi_k \left(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T\right) - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^T,$$

where $\boldsymbol{\Sigma_k} = \text{diag}(\mu_{ki}(1 - \mu_{ki}))$.

Note: Because the covariance matrix cov[x] is no longer diagonal, the mixture distribution can capture correlations between the variables, unlike a single Bernoulli distribution.

Problem 4 (30 Points): Cross-Validation on Classification Models

In this problem, you will implement cross-validation to perform hyperparameter tuning for classification models. Particularly, you will carry out a 10-cross validation on 4 classification models: L_2-regularized Logistic Regression (L2-LR), SVM with linear kernel, SVM with RBF kernel, and Multi-layer Perceptron (MLP). For how to use these models in sklearn, please refer to:

`https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`

`https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

`https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`

In the problem, you treat C in L2-LR and SVM, and number of hidden units (i.e., hidden_layer_sizes) in MLP as the hyperparameter, respectively. To begin with, you need to predefine a series hyperparameter values, e.g., $C = \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ and hidden_layer_sizes $= \{1/10, 1/5, 1/2, 1, 2, 5, 10\} \times$ data features (If hidden_layer_sizes < 1, ignore it). To implement 10-fold cross-validation, you need to first split each dataset into a training set and a testing set, with a ratio 6:4. Then, you need to further split the training dataset into 10 folds, with 9 folds for training and the remaining 1 fold for validation. You will test cross-validation on three datasets: **Iris, Breast cancer, and The 20 newsgroups text dataset**. The details of these dataset can be found in `https://scikit-learn.org/stable/datasets.html`

Please report the mean accuracy of each hyperparameter value on 10 folds and choose the optimal one in each dataset (**10 points per dataset**). Based on the mean accuracy, which of the above classifiers is the best and which is the worse?

(Note: Please do not use the inherent cross validation function, but implement it by yourself)

Problem 5 (10 Points): Reconstructing Europe via MDS

Distances between 24 European cities have been attached as `cities.csv`. Using multi-dimensional scaling (MDS), embed the points on a 2D plane so that you approximately reconstruct the map of Europe. (You may get the map up to a reflection/rotation, so you may have to post-process the answer that you get from MDS.)

Problem 6 (20 Points): The (Random) Walking Dead

An undead walker initially appears at the (0, 0) point of an infinite two-dimensional grid. The walker takes unit (± 1) steps either horizontally or vertically on the grid at random. After N such steps, the walker finds himself at location (x_N, y_N).

1. **5 Points.** Fix $N = 50$ and plot the traces of 3 independent realizations of such a random walk.
2. **7 Points.** We are interested in understanding how far the walker can stray from the origin, i.e., the expected value of the quantity $d = \sqrt{x_N^2 + y_N^2}$. Execute this random walk 10,000 times, and measure the value of d in each realization. Estimate the average value of d.
3. **8 Points.** Repeat the above numerical experiment for different values of N ranging from 10 to 500. Show that d scales as $\sqrt{N}$. (Hint: One way to quantitatively display this scaling is to plot $\log N$ versus $\log d$, and measure the slope.)