- **Gradient Descent**

Consider the nonlinear error surface $E(u, v) = (ue^v - 2ve^{-u})^2$. We start at the point $(u, v) = (1, 1)$ and minimize this error using gradient descent in the uv space. Use $\eta = 0.1$ (learning rate, not step size).

4. What is the partial derivative of $E(u, v)$ with respect to u, i.e., $\frac{\partial E}{\partial u}$?

[a] $(ue^v - 2ve^{-u})^2$

[b] $2(ue^v - 2ve^{-u})$

[c] $2(e^v + 2ve^{-u})$

[d] $2(e^v - 2ve^{-u})(ue^v - 2ve^{-u})$

[e] $2(e^v + 2ve^{-u})(ue^v - 2ve^{-u})$

5. How many iterations (among the given choices) does it take for the error $E(u, v)$ to fall below 10^{-14} for the first time? In your programs, make sure to use double precision to get the needed accuracy.

[a] 1

[b] 3

[c] 5

[d] 10

[e] 17

6. After running enough iterations such that the error has just dropped below 10^{-14}, what are the closest values (in Euclidean distance) among the following choices to the final (u, v) you got in Problem 5?

[a] $(1.000, 1.000)$

[b] $(0.713, 0.045)$

[c] $(0.016, 0.112)$

[d] $(-0.083, 0.029)$

[e] $(0.045, 0.024)$

7. Now, we will compare the performance of "coordinate descent." In each iteration, we have two steps along the 2 coordinates. Step 1 is to move only along the u coordinate to reduce the error (assume first-order approximation holds like in gradient descent), and step 2 is to reevaluate and move only along the v coordinate to reduce the error (again, assume first-order approximation holds). Use the same learning rate of $\eta = 0.1$ as we did in gradient descent. What will the error $E(u, v)$ be closest to after 15 full iterations (30 steps)?

[a] 10^{-1}

[b] 10^{-7}

[c] 10^{-14}

[d] 10^{-17}

[e] 10^{-20}

- **Logistic Regression**

In this problem you will create your own target function f (probability in this case) and data set $\mathcal{D}$ to see how Logistic Regression works. For simplicity, we will take f to be a 0/1 probability so y is a deterministic function of $\mathbf{x}$.

Take $d = 2$ so you can visualize the problem, and let $\mathcal{X} = [-1, 1] \times [-1, 1]$ with uniform probability of picking each $\mathbf{x} \in \mathcal{X}$. Choose a line in the plane as the boundary between $f(\mathbf{x}) = 1$ (where y has to be $+1$) and $f(\mathbf{x}) = 0$ (where y has to be -1) by taking two random, uniformly distributed points from $\mathcal{X}$ and taking the line passing through them as the boundary between $y = \pm 1$. Pick $N = 100$ training points at random from $\mathcal{X}$, and evaluate the outputs y_n for each of these points $\mathbf{x}_n$.

Run Logistic Regression with Stochastic Gradient Descent to find g, and estimate E_{out} (the **cross entropy** error) by generating a sufficiently large, separate set of points to evaluate the error. Repeat the experiment for 100 runs with different targets and take the average. Initialize the weight vector of Logistic Regression to all zeros in each run. Stop the algorithm when $\|\mathbf{w}^{(t-1)} - \mathbf{w}^{(t)}\| < 0.01$, where $\mathbf{w}^{(t)}$ denotes the weight vector at the end of epoch t. An epoch is a full pass through the N data points (use a random permutation of $1, 2, \cdots, N$ to present the data points to the algorithm within each epoch, and use different permutations for different epochs). Use a learning rate of 0.01.

8. Which of the following is closest to E_{out} for $N = 100$?

[a] 0.025

[b] 0.050

[c] 0.075

[d] 0.100

[e] 0.125

9. How many epochs does it take on average for Logistic Regression to converge for $N = 100$ using the above initialization and termination rules and the specified learning rate? Pick the value that is closest to your results.

[a] 350

[b] 550

[c] 750

[d] 950

[e] 1750

• Regularization with Weight Decay

In the following problems use the data provided in the files

http://work.caltech.edu/data/in.dta

http://work.caltech.edu/data/out.dta

as a training and test set respectively. Each line of the files corresponds to a two-dimensional input $\mathbf{x} = (x_1, x_2)$, so that $\mathcal{X} = \mathbb{R}^2$, followed by the corresponding label from $\mathcal{Y} = \{-1, 1\}$. We are going to apply Linear Regression with a non-linear transformation for classification. The nonlinear transformation is given by

$$\phi(x_1, x_2) = (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2, |x_1 - x_2|, |x_1 + x_2|).$$

Recall that the classification error is defined as the fraction of misclassified points.

2. Run Linear Regression on the training set after performing the non-linear transformation. What values are closest (in Euclidean distance) to the in-sample and out-of-sample classification errors, respectively?

[a] 0.03, 0.08

[b] 0.03, 0.10

[c] 0.04, 0.09

[d] 0.04, 0.11

[e] 0.05, 0.10

3. Now add weight decay to Linear Regression, that is, add the term $\frac{\lambda}{N}\sum_{i=0}^{7} w_i^2$ to the squared in-sample error, using $\lambda = 10^k$. What are the closest values to the in-sample and out-of-sample classification errors, respectively, for $k = -3$? Recall that the solution for Linear Regression with Weight Decay was derived in class.

[a] 0.01, 0.02

[b] 0.02, 0.04

[c] 0.02, 0.06

[d] 0.03, 0.08

[e] 0.03, 0.10

4. Now, use $k = 3$. What are the closest values to the new in-sample and out-of-sample classification errors, respectively?

[a] 0.2, 0.2

[b] 0.2, 0.3

[c] 0.3, 0.3

[d] 0.3, 0.4

[e] 0.4, 0.4

5. What value of k, among the following choices, achieves the smallest out-of-sample classification error?

[a] 2

[b] 1

[c] 0

[d] -1

[e] -2

6. What value is closest to the minimum out-of-sample classification error achieved by varying k (limiting k to integer values)?

[a] 0.04

[b] 0.06

[c] 0.08

[d] 0.10

[e] 0.12

- **Neural Networks**

8. A fully connected Neural Network has $L = 2$; $d^{(0)} = 5$, $d^{(1)} = 3$, $d^{(2)} = 1$. If only products of the form $w_{ij}^{(l)} x_i^{(l-1)}$, $w_{ij}^{(l)} \delta_j^{(l)}$, and $x_i^{(l-1)} \delta_j^{(l)}$ count as operations (even for $x_0^{(l-1)} = 1$), without counting anything else, which of the following is the closest to the total number of operations in a single iteration of backpropagation (using SGD on one data point)?

[a] 30

[b] 35

[c] 40

[d] 45

[e] 50

Let us call every 'node' in a Neural Network a unit, whether that unit is an input variable or a neuron in one of the layers. Consider a Neural Network that has 10 input units (the constant $x_0^{(0)}$ is counted here as a unit), one output unit, and 36 hidden units (each $x_0^{(l)}$ is also counted as a unit). The hidden units can be arranged in any number of layers $l = 1, \cdots, L-1$, and each layer is fully connected to the layer above it.

9. What is the minimum possible number of weights that such a network can have?

[a] 46

[b] 47

[c] 56

[d] 57

[e] 58

10. What is the maximum possible number of weights that such a network can have?

[a] 386

[b] 493

[c] 494

[d] 509

[e] 510