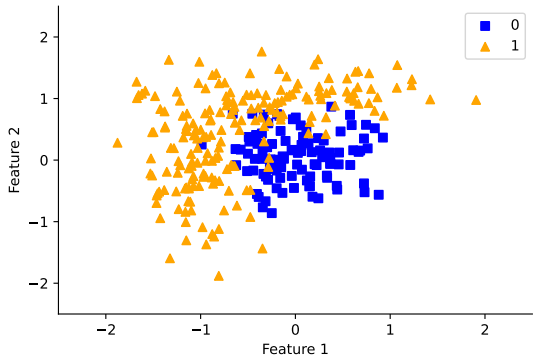
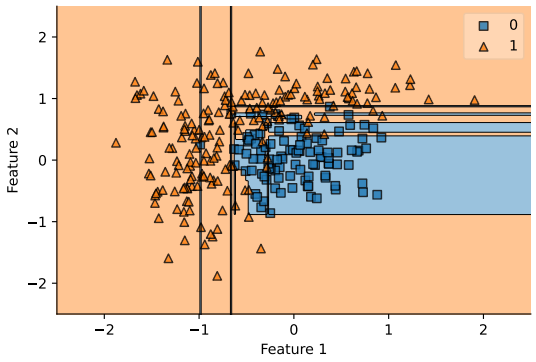
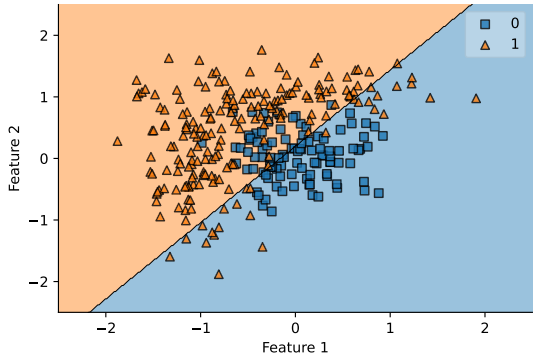
Data 102 (Fall 2022) **Names:**
Discussion Worksheet 7
10/14/2022

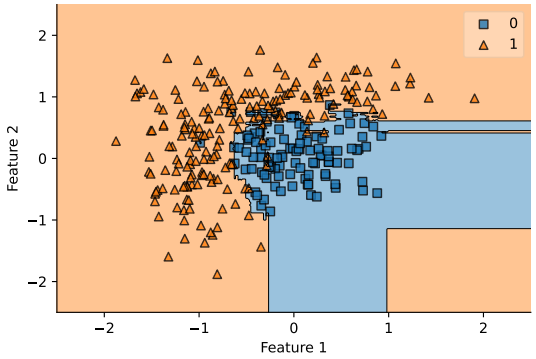
---

1. (a) The four images below show a classification data set, along with the decision boundary for each of three models: (1) logistic regression, (2) a decision tree with no depth limit, and (3) a random forest with no depth limit. Match each model to the corresponding image.









(b) Recall the bias-variance decomposition for the squared error:

$$E[(\hat{y}(x) - y)^2] = \underbrace{E\left[(\hat{y}(x) - E[\hat{y}(x)])^2\right]}_{\text{Variance of prediction } \hat{y}(x)} + \underbrace{(E[\hat{y}(x)] - y)^2}_{\text{Bias}^2 \text{ of } \hat{y}(x)}.$$

Circle the correct word for each sentence:

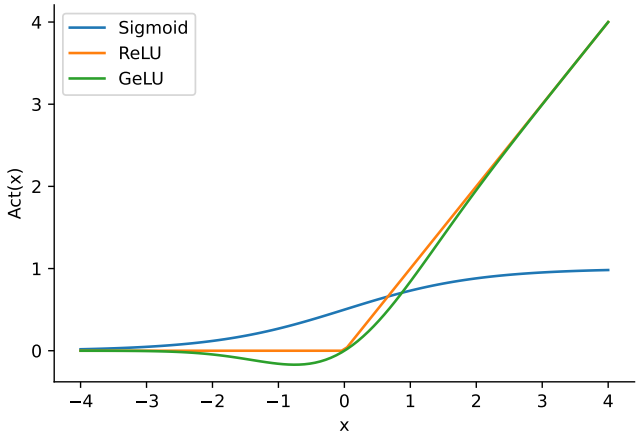
- The bias of logistic regession is (LARGER THAN / SMALLER THAN / EQUAL TO) that of the decision tree.
- The bias of the decision tree is (LARGER THAN / SMALLER THAN / EQUAL TO) that of the random forest model.
- The variance of the decision tree is (LARGER THAN / SMALLER THAN / EQUAL TO) that of the random forest model.

2. We saw in lecture that neural networks can learn complex, non-linear patterns from data using activation functions. One simple activation function which you've seen in this class before is the *sigmoid* function used in Logistic Regression, defined as:

$$\sigma(x) = \frac{1}{1 + \exp(-x))}$$

However, most modern-day implementations of neural networks avoid using sigmoid functions for the nonlinearity, and instead favor functions like the ReLU (*Rectified Linear Unit*) or GELU. The ReLU function is defined as $\text{ReLU}(x) = \max(0, x)$, and the GELU has a formula involving the CDF of a Gaussian (not given here).

We can visualize what these activation functions look like in the plot below:



(a) Referencing the plot above, sketch the derivatives of the sigmoid and ReLU activation functions, overlaid on the same plot.

(b) What happens to the derivative for each of the activation functions as the input gets large?

(c) When optimizing neural networks, why is it bad for the gradient values to be (close to) 0? Why do ReLU and GeLU do a better job than sigmoid of avoiding this problem?

3. Consider a two-layer neural network that computes a real-valued function of the form

$$f_{W_1,w_2}(x) = w_2^T \sigma(W_1^T x)$$

where $x \in \mathbb{R}^m$, $W_1 \in \mathbb{R}^{m \times h}$, $w_2 \in \mathbb{R}^h$, and

$$\sigma(x) = 1/(1 + \exp(-x)).$$

The subscript notation in f_{W_1,w_2} just indicates that W_1 and w_2 are parameters of the function f.

Suppose the neural network uses squared-error prediction loss for data points (x, y):

$$\mathcal{L}(y, f_{W_1,w_2}(x)) = (y - f_{W_1,w_2}(x))^2. \tag{1}$$

(a) For this loss function, draw the corresponding computation graph. (*Hint:* There are four intermediate nodes in the graph, which you should label $z_1, \ldots, z_4$.)

The trick to coming up with computation graphs is to "unroll" the expression into a series of operations on different quantities, starting from the innermost nested expression.

(b) Using the chain rule, write down an expression for $\frac{\partial \mathcal{L}(W_1,w_2)}{\partial w_2}$. Simplify your answer in terms of the intermediate nodes $z_1, \ldots, z_4$.

(c) Using the chain rule, write down an expression for $\frac{\partial \mathcal{L}(W_1,w_2)}{\partial W_1}$. As in the previous part, simplify your answer in terms of the z_i as much as possible.

(*Hint 1:* In addition to the z_i, your answer will involve x, w_2, and the derivative of the sigmoid function.)

(*Hint 2:* If you are having trouble with the vector and matrix notation, start by assuming that w_2 and W_1 are both one-dimensional real numbers.)

(d) Explain why simplifying the derivatives in terms of the z_i allows us to save computation.

Feedback Form

On a scale of 1-5, where 1 = much too slow and 5 = much too fast, how was the pace of the discussion section?

1 2 3 4 5

Which problem(s) did you find most useful?

Which were least useful?

Any other feedback?