Machine Learning

Fall 2024

Name:

Machine Learning as Optimization

Quality Assessed Deliverable

Please fill out this assessment using no external resources, just that awesome brain of yours. If possible, please turn off your phone notifications. If you need a calculator, you can use your phone. Please place it on top of your table and use it there in plain sight.

There are 150 total possible points on this deliverable.

1. [14 points] Below are several examples of machine learning tasks. For each example, specify whether it is an instance of **supervised regression, supervised classification,** or **unsupervised learning**. Write the appropriate letters below.
    a. **Spam Detection**: Classifying emails as "spam" or "not spam" based on labeled email datasets.
    b. **Customer Segmentation**: Grouping customers into distinct segments based on purchasing behavior without predefined labels.
    c. **Stock Market Clustering**: Identifying patterns in stock price movements to group similar stocks without prior labels.
    d. **Predicting House Prices**: Using historical data of house features (size, number of rooms, location) and their sale prices to predict the price of a new house.
    e. **Handwriting Recognition**: Recognizing handwritten digits (0-9) from images with corresponding labels.
    f. **Anomaly Detection**: Detecting unusual credit card transactions that deviate from normal patterns, without labeled examples of fraud.
    g. **Predicting Temperature**: Estimating future temperatures based on historical weather data.

	Letters from above
Supervised Regression	
Supervised Classification	

Unsupervised Learning	

2. [16 points] Explain the components of a supervised learning problem, including the roles of X's, the y's, the model, and the loss function. (Feel free to draw a diagram.) Distinguish between training and testing where appropriate. Use an example from the previous exercise, giving an example of potential X's, y's, model, and loss function.

3. [7 points] Describe the purpose of splitting data into training, validation, and testing sets. Why is each set important?

4. [9 points] Match each loss function to its equation by drawing lines between them.

Cross-entropy	$\frac{1}{N}\sum_{i=1}^{N}(w^T x_i - y_i)^2$
Binary cross-entropy or log loss	$\frac{-1}{N}\sum_{i=1}^{N}[y_i ln(p_i) + (1-y_i) ln(1-p_i)]$
Squared loss	$\sum_{i=1}^{k} - \mathrm{I}[y = i] l(\ln ln \hat{y}_i)$

5. [12 points] In the case of a binary classifier that outputs probabilities:
   a. If a classifier has no clear idea of whether the output for a particular input is 1 or 0, what probability should the classifier output?

   b. If a classifier is relatively certain that the output for a particular input is 1, what probability should the classifier output?

   c. If a classifier is relatively certain that the output for a particular input is 0, what probability should the classifier output?

   d. Plug into the log loss equation for a situation in which the classifier outputted 0.7 and the correct answer was 0. No need to find the actual values, just input these to values into the equation.

6. [20 points] For the model described by the PyTorch code below:

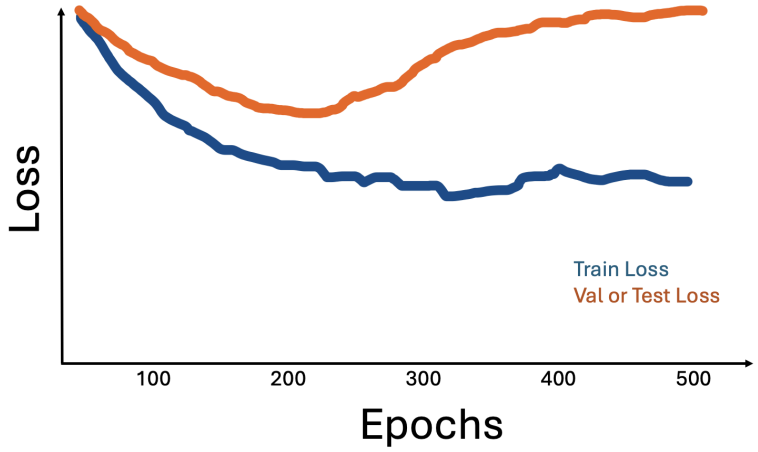
```
class DrawMe(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(DrawMe, self).__init__()
        self.linear_1 = nn.Linear(3, 2)
        self.linear_2 = nn.Linear(2, 1)

    def forward(self, x):
        out = self.linear_1(x)
        out = F.sigmoid(out)
        out = self.linear_2(out)
        # Leave this last line out because our loss function takes care of it.
        # Including it here for clarity.
        # out = F.sigmoid(out, dim=1)
        return out
```

   a. Describe / Draw the model

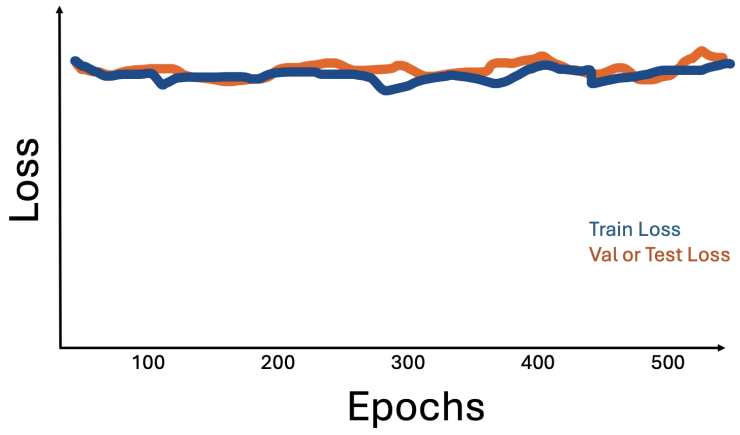
   b. What would be an appropriate loss function for this model?

7. [20 points] What does each of these loss plots suggest might be going on with the multilayer perceptron model? Name at least two interventions you might use for each of these situation (e.g., add another layer).

a.



b.



8. [15 points] Draw a dataflow diagram and calculate the partial derivatives with respect to the variables x, y, and z for this following expression: $x^2(xy + z)$

9. [25 points] Describe how partial derivatives are used in learning for neural networks. What is this process called? Name at least 3 things that change because of this process. Name one thing that stays the same.

10. [12 points] Rank these PyTorch models based on the number of parameters.

Highest number of parameters: Model _____

Middle number of parameters: Model _____

Smallest number of parameters: Model _____

```
class Model_A(nn.Module):
    def __init__(self):
        super(Model_A, self).__init__()
        self.fc1 = nn.Linear(10, 64)
        self.out = nn.Linear(64, 4)

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = self.out(x)
        return x
```

```
class Model_B(nn.Module):
    def __init__(self):
        super(Model_B, self).__init__()
        self.fc1 = nn.Linear(10, 128)
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 32)
        self.fc4 = nn.Linear(32, 16)
        self.fc5 = nn.Linear(16, 8)
        self.out = nn.Linear(8, 1)

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = F.relu(self.fc3(x))
        x = F.relu(self.fc4(x))
        x = F.relu(self.fc5(x))
        x = self.out(x)
        return x
```

```
class Model_C(nn.Module):
    def __init__(self):
        super(Model_C, self).__init__()
        self.fc1 = nn.Linear(10, 16)
        self.fc2 = nn.Linear(16, 10)
        self.out = nn.Linear(10, 4)

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.out(x)
        return x
```

Machine Learning

Fall 2024