Exercise Week 47

Isak O. Rukan, Edvard B. Rørnes

November 18, 2024

Exercise 1: Linear and Logistic Regression Methods

1. **What is the main difference between ordinary least squares and Ridge regression?**
   OLS minimizes the residual sum of squares without any penalty. Ridge regression adds a regularization term proportional to the square of the model coefficients, $C_{\text{OLS}} + \lambda \boldsymbol{\beta}_i^2$.

2. **Which kind of dataset would you use logistic regression for?**
   For datasets where the dependent variable is binary or categorical, such as classification problems.

3. **In linear regression you assume that your output is described by a continuous non-stochastic function $f(x)$. Which is the equivalent function in logistic regression?**
   The sigmoid function:
   $$f(x) = \frac{1}{1 + e^{-z}},$$
   where $z = \sum_{i=0}^{n} \beta_i x_i$.

4. **Can you find an analytic solution to a logistic regression type of problem?**
   No, there is no closed-form analytic solution because the logistic regression loss function is not linear.

5. **What kind of cost function would you use in logistic regression?**
   We would use the cross-entropy cost function:
   $$C = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right],$$

where p_i is the predicted probability for the i-th sample.

Exercise 2: Deep Learning

1. **What is an activation function and discuss the use of an activation function? Explain three different types of activation functions?**
   An activation function introduces non-linearity to the neural network, allowing it to learn complex patterns.

   - Sigmoid: Outputs values in $[0, 1]$; used in binary classification.
   - ReLU: Outputs $f(x) = \max(0, x)$; efficient but may cause "dead neurons."
   - Leaky ReLU: A variant of ReLU, $f(x) = \max(a, x), a \ll 1$ with small slopes for negative inputs, solving the "dead neuron" issue.

2. **Describe the architecture of a typical feedforward Neural Network (NN).**
   An FFNN is made up of an input layer, hidden layers, and an output layer. Information flows in one direction, and each layer consists of neurons connected to other neurons via weights.

3. **You are using a deep neural network for a prediction task. After training your model, you notice that it is strongly overfitting the training set and that the performance on the test isn't good. What can you do to reduce overfitting?**
   Techniques we considered were simply L1/L2 regularization, use more data, reduce model complexity and use cross-validation.

4. **How would you know if your model is suffering from the problem of exploding Gradients?**
   Look for `nan`, large (increasing) loss or just simply check the gradient during training.

5. **Can you name and explain a few hyperparameters used for training a neural network?**
   Learning rate, batch size, number of layers, number of neurons per layer, activation functions, regularization strength, and optimizer type.

6. **Describe the architecture of a typical Convolutional Neural Network (CNN).**

Made up of convolutional layers, pooling layers, and fully connected layers.

7. **What is the vanishing gradient problem in Neural Networks and how to fix it?**
   Occurs when gradients become too small to update weights effectively. To solve one may use activation functions like ReLU or Leaky ReLU, or use techniques like batch normalization.

8. **When it comes to training an artificial neural network, what could the reason be for why the cost/loss doesn't decrease in a few epochs?**
   Learning rate too high/low, vanishing/exploding gradients, insufficient epochs.

9. **How does L1/L2 regularization affect a neural network?**
   L1 regularization sets small certain weights to zero, and penalizes large coefficients. L2 regularization prevents overfitting by penalizing squared weights.

10. **What is (are) the advantage(s) of deep learning over traditional methods like linear regression or logistic regression?**
    Handles high-dimensional, unstructured data. Automatically learns complex patterns and hierarchical representations. Overall faster than closer to analytical methods.

Exercise 3: Decision Trees and Ensemble Methods

1. **Mention some pros and cons when using decision trees.**
   Pros: Simple to interpret, handles categorical/continuous data, requires little preprocessing.
   Cons: Prone to overfitting, sensitive to small data changes.

2. **How do we grow a tree? And which are the main parameters?**
   To grow a decision tree, you split the data into smaller groups step by step. At each step a rule is chosen to separate the data into two parts. This process continues until the groups are small enough or meet a stopping rule. The main parameters are:

   - Maximum depth: How many levels the tree can have.

- Minimum samples per leaf: The smallest number of data points allowed in each group.
- Minimum samples to split: The smallest number of data points needed to make a split.

3. **Mention some of the benefits with using ensemble methods (like bagging, random forests and boosting methods)?**
   Ensemble methods help reduce overfitting and variance.

4. **Why would you prefer a random forest instead of using Bagging to grow a forest?**
   Random forest uses Bagging to create multiple different trees and then proceeds to randomly selects a subset of the features at each split for each tree. This can prevent individual trees from relying too much on dominant features and ensures that all features have a chance to contribute across the forest.

5. **What is the basic philosophy behind boosting methods?**
   Boosting methods aim to improve performance of models that perform slightly better than random guessing (weak learners) by combining them. They work by creating the next model by analyzing the errors of the previous model. The final model is formed by combining the predictions from all models. How this is weighted is given by the specific boosting method.

Exercise 4: Optimization Part

1. **Which is the basic mathematical root-finding method behind essentially all gradient descent approaches(stochastic and non-stochastic)?**
   Newton's method.

2. **And why don't we use it? Or stated differently, why do we introduce the learning rate as a parameter?**
   Helps with convergence by controlling the step size.

3. **What might happen if you set the momentum hyperparameter too close to 1 (e.g., 0.9999) when using an optimizer for the learning rate?**
   May overshoot the minimum causing instability or divergence.

4. **Why should we use stochastic gradient descent instead of plain gradient descent?**
   Faster updates, better generalization, and reduced computation per step.

5. **Which parameters would you need to tune when use a stochastic gradient descent approach?**
   Learning rate, momentum, batch size, and regularization terms.

Exercise 5: Analysis of Results

1. **How do you assess overfitting and underfitting?**
   Overfitting: Training error low, test error high.
   Underfitting: Both training and test errors high.

2. **Why do we divide the data in test and train and/or eventually validation sets?**
   To evaluate model performance by making sure it has not already seen the data and is interpreting potential noise.

3. **Why would you use resampling methods in the data analysis? Mention some widely popular resampling methods.**
   To improve model robustness and reduce bias/variance.
   Methods: Cross-validation, bootstrapping.