Machine Learning – MTH 366

Q1 a)
How do concepts like regularization, feature scaling, and cross-validation contribute to enhancing model performance, and how do these principles link with various supervised learning algorithms, such as support vector machines given their distinct objectives, underlying assumptions, and prediction mechanisms?

Q1 b)
Consider a binary classification problem using the Naive Bayes algorithm. We have a dataset containing information about the weather conditions (sunny, rainy, or cloudy) and whether people play golf on those days (yes or no). The dataset has the following distribution:

Total days: 10
Days with golf: 6
Days without golf: 4

Weather condition distribution:
Sunny: 5 (4 with golf, 1 without golf)
Rainy: 3 (1 with golf, 2 without golf)
Cloudy: 2 (1 with golf, 1 without golf)

Given a new day with rainy weather, calculate the probability of people playing golf using the Naive Bayes algorithm.

Q2 a)
Consider a simple feedforward neural network with one input layer, one hidden layer, and one output layer. The input layer has 2 neurons, the hidden layer has 3 neurons, and the output layer has a single neuron. The activation function for the hidden layer neurons is Rectified Linear Unit, and for the output layer neuron, it is a sigmoid function.

Given the input vector [0.5, 0.7], the weights matrix from the input to the hidden layer as W1 = [[0.1, 0.2, 0.3], [0.4, 0.5, 0.6]], and the weights matrix from the hidden to the output layer as W2 = [[0.7], [0.8], [0.9]], calculate the output of the neural network.

Q2 b)
Consider a simple convolutional neural network with an input image of size 6x6x3 (height x width x channels) and a single convolutional layer with a 3x3x3 filter/kernel, stride of 2, and no padding.

Calculate the dimensions of the feature map produced by the convolutional layer and the total number of parameters in the filter/kernel, including the bias term.

Q3

Implement logistic regression using a built-in Keras dataset for binary classification. You will preprocess the dataset, build and train the model, evaluate its performance, and visualize the results.

Dataset:

You will use the Keras built-in dataset, the "IMDB movie review dataset" for binary classification. This dataset consists of 50,000 movie reviews from the Internet Movie Database (IMDB) labeled as positive or negative (1 or 0).

Steps:

- Import necessary libraries:
  First, import the required libraries for this task, including TensorFlow, Keras, NumPy, and Matplotlib.

- Load and preprocess the dataset:
  Load the IMDB dataset, and preprocess it by setting the maximum number of words to consider as features (e.g., 10,000). Then, pad the sequences to have the same length (e.g., 100).

- Build the logistic regression model:
  Create a simple logistic regression model using the Keras Sequential API. The model will consist of an Embedding layer, a GlobalAveragePooling1D layer, and a Dense layer with sigmoid activation for binary classification.

- Compile the model:
  Compile the model using the binary cross-entropy loss, the Adam optimizer, and the accuracy metric.

- Train the model:
  Train the model for a number of epochs (e.g., 20) using the training dataset and a validation split (e.g., 0.2).

- Evaluate the model:
  Evaluate the model's performance on the test dataset.

- Visualize the results:
  Plot the training and validation accuracy and loss over time.

After completing these steps, you will have successfully implemented logistic regression on the Keras built-in IMDB movie review dataset for binary classification.

Q4

Implement a convolutional neural network (CNN) using the built-in Keras Fashion-MNIST dataset for image classification. You will preprocess the dataset, build and train the CNN model, evaluate its performance, and visualize the results.

Dataset:

You will use the Keras built-in dataset, the "Fashion-MNIST" dataset for image classification. This dataset consists of 70,000 28x28 grayscale images in 10 classes, with 7,000 images per class. There are 60,000 training images and 10,000 test images.

Steps:

- Import necessary libraries:
  First, import the required libraries for this task, including TensorFlow, Keras, NumPy, and Matplotlib.

- Load and preprocess the dataset:
  Load the Fashion-MNIST dataset, and preprocess it by normalizing the pixel values to be in the range of 0 to 1.

- Build the CNN model:
  Create a CNN model using the Keras Sequential API. The model will consist of several convolutional layers with ReLU activation, max-pooling layers, a flattening layer, one or more fully connected layers with ReLU activation, and an output layer with softmax activation for multi-class classification.

- Compile the model:
  Compile the model using the categorical cross-entropy loss, an appropriate optimizer (e.g., Adam or RMSprop), and the accuracy metric.

- Train the model:
  Train the model for a number of epochs (e.g., 30) using the training dataset and a validation split (e.g., 0.1).

- Evaluate the model:
  Evaluate the model's performance on the test dataset.

- Visualize the results:
  Plot the training and validation accuracy and loss over time.

After completing these steps, you will have successfully implemented a convolutional neural network (CNN) on the Keras built-in Fashion-MNIST dataset for image classification.