Jimmy Wallace

Dr. Fosler-Lussier

CSE 5526 Introduction to Neural Networks

February 24, 2020

Lab 2 Radial Basis Function Neural Networks

Introduction:

The purpose of this lab is to gain familiarity with implementing radial basis function neural networks. Radial basis function neural networks are two-layered neural networks that use a basis function to compute the outputs of the hidden nodes. Furthermore, radial basis function neural networks implement clustering algorithms and linear regression algorithms to train its weights and biases. By implementing a radial basis function neural network, we will gain experience with clustering and linear regression. Also, we will construct multiple different radial basis function neural networks using a varied number of bases and learning rates to determine the effect of each hyperparameter respectively.

Procedure:

In this lab, we implemented a radial basis function neural network with Gaussian basis functions that has a single input scalar and a single output scalar. The Gaussian basis function can be seen below.

The parameters of the Gaussian basis function are the input vector, the centroid of the given cluster, and the variance of the given cluster respectively. The variance parameter of the Gaussian basis can either be held constant for simplicity or can be computed as the variance of the respective cluster. In this lab, we used the intra-cluster variance of each cluster as the variance of each Gaussian basis as well as a constant variance given by the equation listed below.

By using intra-cluster variance as well as a constant variance, we are able to determine the efficacy of each chose of variance.

We trained the radial basis function neural networks to model the sampling function with added uniform noise seen below.

The uniformly distributed noise was sampled over the range [-0.1, 0.1]. We trained this neural network on a training sample of 75 uniformly distributed training samples over the range [0, 1]. To train the radial basis function neural network, we implemented the k-means algorithm to train the hidden layer nodes and we implemented multivariate linear regression with a least mean squares weight update and bias update rules to train the output layer nodes. We trained the radial basis function neural networks for a total of 100 epochs, or until convergence, and then compared the efficacy of each trained model with the original training data with a sum of squared errors cost function. We repeated this training process ten times with a varying number of bases and learning rates to determine the ideal hyperparameters of the radial basis function neural network according to the sum of squared errors cost function for an intra-cluster variance Gaussian basis function. Then, we repeated this training process with the original training data with a constant variance for the Gaussian basis function

Results:

In this lab, we constructed a total of twenty different radial basis function neural networks to model the sampling function h with added uniformly distributed noise. Each radial basis function neural network was implemented with a different combination of hyperparameters. The hyperparameters of interest in our case are the number of bases, the learning rate, and variance. The variance is a hyperparameter because we can either use the intra-cluster variance of each cluster as the variance of each Gaussian basis function respectively or we can use a constant variance for each Gaussian basis function given by the equation above. Below are the twenty radial basis function neural networks and their respective loss as compared to the training data.

A close up of a map

Description automatically generated

A close up of a map

Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generated

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

The twenty above graphs represent the efficacy of each constructed radial basis function neural network according to its given hyperparameters. Each graph includes a legend that defines each model’s respective hyperparameters as well as the model’s cost as compared to the training data. Below is a table that contains the information listed in each graph in tabular form for the radial basis function neural networks that used intra-cluster variance as the variance parameter for each Gaussian basis function.

A screen shot of a computer

Description automatically generated

The table below describes the efficacy of the ten radial basis function neural networks shown above that used a constant variance as the variance parameter for each Gaussian basis function.

A screenshot of a computer

Description automatically generated

Conclusions:

The goal of this lab was to gain familiarity with radial basis function neural networks and their respective training process. As previously mentioned, each radial basis function neural network is constructed given two hyperparameters: the number of bases and the learning rate. An additional goal of this lab was to determine the effects of the hyperparameters on radial basis function neural networks. Derived from the results listed above, one notices that the radial basis function neural network minimizes the error with respect to the training data when the number of bases is equal to 3. However, as the number of bases is increased for each model, the error tends to minimize. This is because as the number of bases is increased, the radial basis function neural network begins to overfit the training set and extrapolate on the provided information. As for the learning rates, there is not much of a noticeable difference between using a learning rate of 0.01 and 0.02 with varied number of bases respectively. As the learning rate hyperparameter increases, the model tends to be more confident in its weight and bias updates and therefore tends to overfit the training data. Furthermore, the overfitting of the training set tends to occur when intra-cluster variance is used as the variance of each Gaussian basis function. The models that use a constant variance as the variance parameter of each Gaussian basis function tended to not overfit the training set because the variance was held constant as to control the width of the Gaussian basis function. According to the models, the error is minimized with respect to the training set when the number of bases is equal to 16, the learning rate is 0.02, and a constant variance hyperparameter is utilized. It is important to note that as the number of bases increased for the intra-cluster variance models, the model tended to overfit the data and extrapolate. However, for the constant variance models, as the number of bases increased, the model tended to not overfit the data and minimize the error function with respect to the training data. This lab allowed students to experiment with training radial basis functions and visually see the effects of a model overfitting a training set. This is important because when training any neural network, it is essential to choose the optimal parameters that minimize the error with respect to the training set while also not overfitting the training set.