Approximating a Two-Dimensional Gaussian Density Function Using Artificial Neural Networks

Sami Taider
Student Number: 84401097

1 Introduction

In this assignment, we used ANNs to approximate a two-dimensional Gaussian probability density function (PDF) with given parameters. Our goal was to design and train a model capable of accurately estimating the PDF, testing its predictions against true values.

Gaussian Distribution Parameters:

- **Mean**: [0, 0]
- **Covariance Matrix**: $\begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix}$

2 Multi-Dimensional Gaussian Distribution

The Gaussian (normal) distribution in two dimensions is represented by an ellipsoidal contour plot, where the spread and orientation are defined by the mean and covariance matrix. The given covariance matrix defines both the spread and correlation between the variables x_1 and x_2, making this an ideal case for testing ANN approximation capabilities in capturing intricate data distributions.

3 Data Generation

For training, we generated 1,000 samples from a multivariate normal distribution with the specified mean and covariance matrix, representing the target 2D Gaussian distribution. Each data point, X_{train}, was assigned a corresponding PDF value y_{train}, calculated using the Gaussian PDF.

4 ANN Architecture

We designed an ANN with the following architecture:

- **Input Layer**: 2 neurons (corresponding to x_1 and x_2)
- **Hidden Layers**: 128 neurons with **ReLu** activation
- **Output Layer**: 1 neuron (predicting the PDF value)

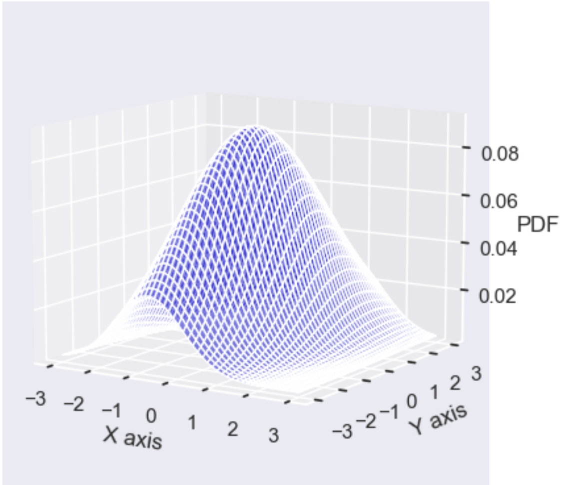

Figure 1: 3D plot of generated training data from a Gaussian distribution with mean [0, 0] and covariance matrix $\begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix}$.

This architecture was then iteratively refined to improve performance. Variations included adding more hidden layers, increasing neurons, experimenting with activation functions, and adjusting the learning rate.

4.1 Architecture Enhancements

- **Additional Layers**: We tested one to two additional hidden layers with **ReLU** activation functions.
- **Different Activation Functions**: ReLU provided faster convergence and better approximation compared to Tanh and Sigmoid in some configurations.

5 Training the Model

Hyperparameters:

- **Learning Rate**: Set to 0.01
- **Loss Function**: Mean Squared Error (MSE)
- **Optimizer**: Adam

We trained the model over 5,000 epochs, tracking the loss to evaluate convergence. MSE loss provided a clear metric for model performance, and by plotting loss over epochs, we could observe learning stability and accuracy improvements.

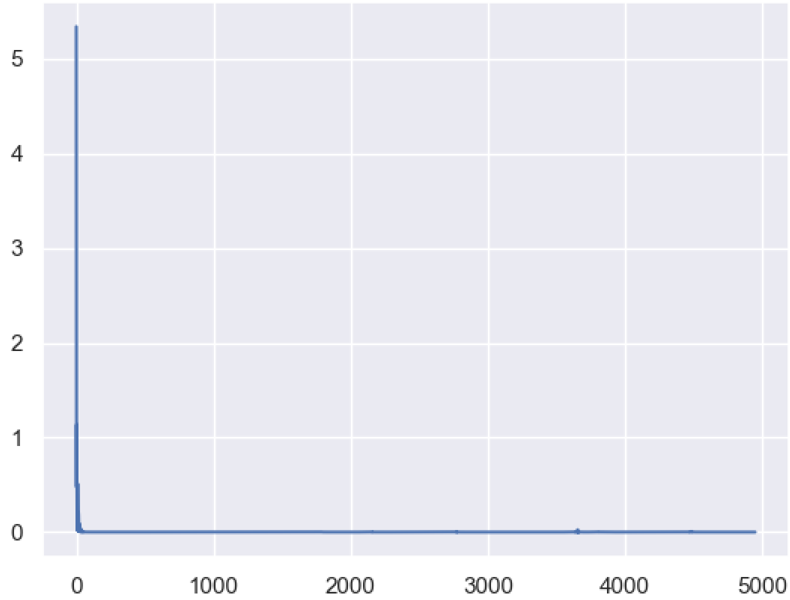

Figure 2: Training loss across epochs, showing model convergence and stability over time.

6 Results and Analysis

6.1 Model Predictions vs. True PDF

To evaluate the model's accuracy, we generated a test grid of points across x_1 and x_2, calculated the true PDF values, and compared these with the model's predictions.

6.2 Section at $y = 0$

A critical evaluation step was to take a section at $y = 0$ and compare the predicted vs. true PDF values along this slice. This visualization provided a direct comparison of the ANN's ability to replicate the PDF shape along a specific line in the distribution.

6.3 Contour Plots

Contour plots offer a comprehensive view of the ANN's ability to capture the shape of the distribution. Here, we compared true PDF contours with ANN-predicted contours.

7 Experiments with Alternate Architectures

To assess potential performance gains, we experimented with:

- **Two additional hidden layers**: Improved approximation in high-density areas.
- **ReLU and LeakyReLU activation**: Improved convergence rates and overall accuracy.

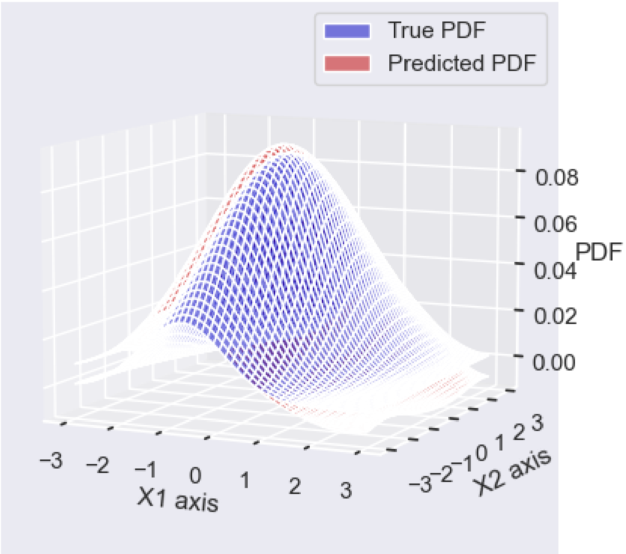

Figure 3: 3D plot of predicted PDF surface (in red) overlaid on true PDF surface (in blue), showing approximation accuracy.

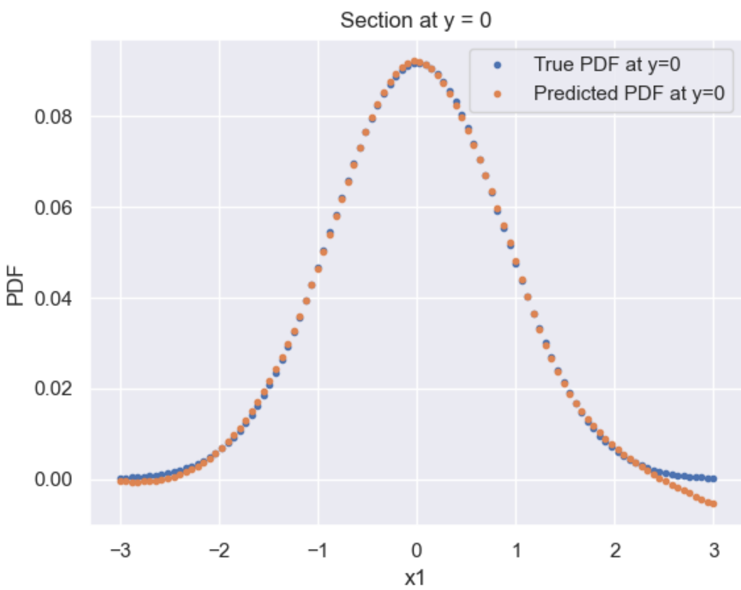

Figure 4: Comparison of true and predicted PDF values at $y = 0$.

8 Challenges and Learnings

Challenges:

- **Training Stability**: Higher complexity models led to stability issues, requiring careful adjustment of learning rates and optimization parameters.
- **Approximation Accuracy**: Accurately capturing the distribution's correlation

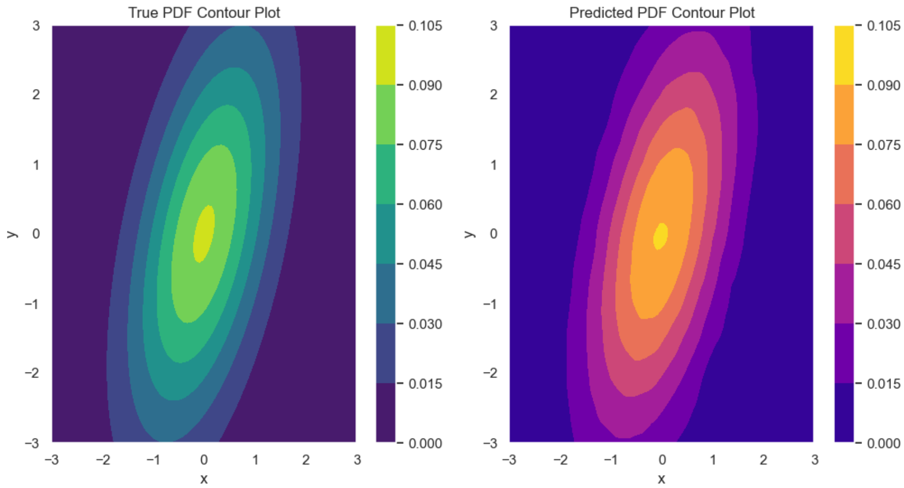

Figure 5: Comparison of true and predicted PDF contour plots.

required tuning layer configurations and balancing model complexity.

Learnings: This task demonstrated the importance of balancing model complexity with stability. We gained insights into effective architecture choices for density function approximation and the impact of activation functions on convergence.

9 Conclusion

The ANN successfully approximated the target 2D Gaussian distribution, with refined architectures showing notable improvements in accuracy. Experimentation revealed the impact of architectural choices on the model's ability to generalize and underscored the importance of activation functions and layer configuration for similar tasks.