https://doi.org/10.31891/2307-5732-2025-355-96
УДК 004.032.26:519.2

SHKURAT OKSANA
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»
https://orcid.org/0000-0001-7633-9121
e-mail: shkurat@pzks.fpm.kpi.ua
FEDORCHUK IVAN
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»
https://orcid.org/0009-0009-4429-1683
e-mail: fedorchuk.kp93@gmail.com

ALGORITHMIC AND SOFTWARE METHOD FOR PREDICTING DATA WITH MULTIMODAL DISTRIBUTION BASED ON THE MDN MODEL

The development of machine learning models capable of accurately predicting data with multimodal distributions is a critical direction in modern data analysis. Such data commonly arise in practical applications where a single input can correspond to multiple valid outputs, for example, in robotic control systems, image processing, or pattern recognition. Traditional neural network models, which rely on deterministic prediction strategies, are often limited in their ability to capture this variability and uncertainty. To address this limitation, this paper presents a modified algorithmic and software method based on the Mixture Density Network (MDN), incorporating a probabilistic method into the loss function calculation during training. Applying a numerical integral into the proposed probabilistic method, making the model more stable and interpretable during training. The study includes a comparative analysis of the classical MDN and the proposed method using synthetic datasets with clearly defined multimodal characteristics, as well as a real-world simulation of a robotic arm positioning task, where multiple angle configurations can achieve the same target coordinates. Additional complexity is introduced by modeling the simultaneous operation of two robotic arms, further emphasizing the model's capacity to resolve multiple overlapping outcomes. The experimental results demonstrate that the modified MDN achieves a consistent reduction in prediction error and training loss across all test scenarios, outperforming both the original MDN and a conventional least-squares method. Despite an increase in training time, the computational efficiency of the final model remains unaffected. These findings highlight the practical relevance and scalability of the proposed method for improving prediction accuracy in complex multimodal systems, offering valuable potential for broader applications in intelligent automation and decision-making systems.

Keywords: data with multimodal distribution, Gaussian distribution, MDN model, loss function.

ШКУРАТ ОКСАНА
ФЕДОРЧУК ІВАН
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

АЛГОРИТМІЧНО-ПРОГРАМНИЙ МЕТОД ДЛЯ ПРОГНОЗУВАННЯ ДАНИХ З МУЛЬТИМОДАЛЬНИМ РОЗПОДІЛОМ НА ОСНОВІ МОДЕЛІ MDN

У даній роботі розглянуто задачу розроблення методу прогнозування мультимодальних даних на основі технології машинного навчання для покращення точності прогнозу. Проведений аналіз існуючих методів прогнозування мультимодальних даних, зокрема методу машинного навчання на основі моделі MDN та методу найменших квадратів. Запропоновано програмний метод прогнозування мультимодальних даних на основі імовірнісної моделі машинного навчання архітектури MDN, що демонструє збільшення точності прогнозу.

Ключові слова: дані з мультимодальним розподілом, розподіл Гауса, модель MDN, функція втрат.

Стаття надійшла до редакції / Received 06.05.2025
Прийнята до друку / Accepted 06.06.2025

Introduction

Likelihood models play a pivotal role in contemporary machine learning, especially in probabilistic modeling and statistical inference. These models provide a rigorous mathematical framework for extracting meaningful insights from data, particularly under conditions of uncertainty or incomplete information. They are particularly advantageous in scenarios where a single input may logically correspond to multiple correct outputs – situations common in many practical applications, such as image processing, pattern recognition, and decision-making tasks.

One illustrative example is semantic image segmentation, where the primary objective is to assign each pixel of an image to a specific predefined class. This process becomes particularly challenging near boundary regions, where pixels often represent transitional or overlapping areas between distinct classes, leading to significant classification ambiguity. Such uncertainty is heightened when pixels might plausibly belong to multiple categories simultaneously [1] due to overlapping object characteristics or the inherent complexity of the scene. Data exhibiting this kind of uncertainty typically display a multimodal distribution, making it difficult to achieve accurate classification with conventional deterministic models.

To handle such multimodal data, likelihood-based approaches [2] have been successfully employed, since they offer predictions in terms of probabilities, assigning likelihoods to various possible outcomes rather than limiting the prediction to a single deterministic result. Specifically, the Mixture Density Network (MDN) has emerged as one of the most effective likelihood-based neural network frameworks [3]. MDN utilizes a mixture of Gaussian

distributions to estimate the likelihoods of multiple potential outcomes, providing flexibility to represent complex, multimodal output spaces effectively. The number of Gaussian components – an important hyperparameter – determines the flexibility and capacity of MDNs to model diverse data distributions accurately.

Despite the success of MDNs in representing multimodal data, its standard implementation relies predominantly on likelihood-based predictions, which may limit performance in certain complex scenarios, especially when dealing with distinctly bimodal or sharply multimodal data distributions. To address this limitation, this paper investigates a modification of the MDN framework by integrating probabilistic modeling techniques to enhance the learning capability and accuracy of MDNs, specifically tailored for accurately describing data with bimodal and multimodal distribution. The proposed probabilistic extension is designed to enable better differentiation among multiple valid outputs, thereby improving the precision and robustness of predictions in challenging multimodal classification tasks.

Analysis of related research

Modeling multimodal data – where a single input may correspond to multiple valid outputs – presents significant challenges in machine learning. Traditional deterministic models often fall short in capturing the inherent uncertainty and variability of such data. To address this, researchers have explored various probabilistic modeling approaches that can effectively represent and predict multimodal distributions.

The classical MDN, introduced by Bishop [3], combines neural networks with a mixture of Gaussian distributions to model conditional probability densities. While effective in capturing multimodal outputs, MDNs rely on the negative log-likelihood (NLL) as a loss function, which can lead to numerical instability, especially when the predicted probabilities are extremely low or high. This instability can hinder the training process and affect the model's predictive performance. Recent research has proposed alternative loss functions to better handle multimodal regression tasks [4]. For instance, the use of multi-bin loss functions has been suggested to address the limitations of L2 loss, which assumes a unimodal Gaussian distribution and often leads to blurred predictions in multimodal scenarios. By discretizing the output space into multiple bins and assigning probabilities to each, models can better capture the diversity of possible outcomes.

Variational methods, such as Variational Autoencoders (VAEs), have been employed to model complex multimodal distributions [5]. VAEs introduce a probabilistic latent space, allowing the model to capture the underlying data distribution more effectively. This approach has been particularly useful in scenarios where the data exhibits high variability and uncertainty.

Incorporating domain knowledge into probabilistic models has led to the development of Physics-guided Mixture Density Networks (PgMDNs) [6]. These models integrate physical laws and constraints into the MDN framework, enhancing the model's ability to predict outcomes that are consistent with known physical behaviors. This approach has shown promise in fields such as engineering and environmental modeling.

Probabilistic models have also been applied to human-robot interaction tasks [7], where understanding and predicting human behavior is crucial. By learning from demonstrations, models can capture the multimodal nature of human actions and improve the robot's ability to interact naturally and effectively with humans.

In autonomous driving and robotics, predicting the trajectory of agents is a multimodal problem due to the multitude of possible future paths. Recent studies have introduced novel loss functions, such as Offroad Loss and Direction Consistency Error, to improve the diversity and accuracy of predicted trajectories [8]. These enhancements enable models to better capture the range of plausible future movements.

Beyond MDNs, multimodal deep learning approaches have been developed to process and integrate information from multiple modalities, such as text, images, and audio. These models leverage the complementary nature of different data types to improve prediction accuracy and robustness. For example, in protein function prediction, integrating sequence and structural information through multimodal models has led to significant performance gains [9].

Study objectives formulation

This study aims to describe a developed algorithm for training a modified Mixture Density Network (MDN) model, which employs a probabilistic approach to calculate the loss function, replacing the likelihood-based method used in the classical MDN. Furthermore, the study seeks to analyze and evaluate the effectiveness of this approach in comparison to other neural network models across various datasets exhibiting multimodal distributions.

Main part of study

First, the general structure of the Mixture Density Network (MDN) and the basic principles of its operation are examined. An MDN integrates a neural network with a Mixture of Gaussians (MoG), a probabilistic model that represents a distribution as a weighted sum of multiple Gaussian (normal) components.

A Mixture of Gaussians (MoG) is a probabilistic model that represents a distribution as a weighted sum of multiple Gaussian (normal) distributions. Mathematically, the probability density function of a MoG is (1):

$$p(y) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y|\mu_k, \sigma_k) \quad (1)$$

In this formula π_k is a weight for the k-th Gaussian so that (2):

$$\sum_{k=1}^{K} \pi_k = 1 \quad (2)$$

K is the number of Gaussian components in the mixture. $\mathcal{N}(y|\mu_k, \sigma_k)$ represents the k-th Gaussian distribution with mean μ_k and covariance σ_k, which can be described by probability density function of Gaussian model (3):

$$\mathcal{N}(y|\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(y-\mu_k)^2}{2\sigma_k^2}} \tag{3}$$

For the continuous variable y its overall cumulative probability over MoG equals 1.

An MDN, as a neural network, typically consists of an input layer, one or more hidden layers, and an output layer. The input layer receives the data, which is then processed through fully connected hidden layers utilizing non-linear activation functions such as ReLU, sigmoid, or tanh. The key distinction between an MDN and conventional shallow or deep neural networks lies in its output layer. Instead of predicting a single deterministic output, the MDN outputs the parameters of a MoG: specifically, the means, covariances, and mixing coefficients for each Gaussian component in the mixture. This enables the MDN to predict a full probability distribution over potential outcomes, effectively capturing uncertainty and offering a range of predictions, each with an associated probability. The number of Gaussian components is treated as a hyperparameter in the model.

Training an MDN involves minimizing a loss function by adjusting the model parameters. Traditionally, the loss function is based on the negative log-likelihood (NLL), which is standard for models that predict probability distributions. For MDN loss function can be expressed by formula (4):

$$L[\phi] = \sum_{i=1}^{I} -\log\left(\sum_{k=1}^{K} \pi_k(x_i, \phi)\mathcal{N}\left(y_i \middle| m_k(x_i, \phi), \sigma_k(x_i, \phi)\right)\right) \tag{4}$$

However, this paper explores the use of a probabilistic approach as an alternative to the likelihood-based method. One motivation for this shift is that the computed likelihood of a specific value y can exceed 1, potentially resulting in a negative loss value. This can compromise the effectiveness of model training and make the results harder to interpret. Ideally, a loss function should converge to 0 when the model perfectly fits the data, serving as a clear indicator for the end of training. This property is not guaranteed when using likelihood-based loss.

It is important to note that the Gaussian model represents probability distributions for continuous variables, implying that we can only compute the probability of a value falling within a specific interval, rather than having an exact value. For MoG to calculate the probability of value y to be in interval $[y_1, y_2]$ following expression (5) is used:

$$p(y_1, y_2) = \int_{y_1}^{y_2} \sum_{k=1}^{K} \pi_k \mathcal{N}(y|\mu_k, \sigma_k)\, dy \tag{5}$$

Therefore, as a modification of the method, we propose using the following loss function (6) instead of the original (4):

$$L[\phi] = \sum_{i=1}^{I} -\log\left(\int_{y_i-\varepsilon}^{y_i+\varepsilon} \sum_{k=1}^{K} \pi_k(x_i, \phi)\mathcal{N}\left(y \middle| m_k(x_i, \phi), \sigma_k(x_i, \phi)\right) dy\right) \tag{6}$$

A notable consequence of adopting the probabilistic approach is an increase in model training time, due to the additional computational effort required. However, the runtime performance of the trained model remains unaffected, since the modification only impacts the loss function used during training. For the numerical approximation of the definite integral involved in the loss calculation, the trapezoidal rule is employed.

As the first task to evaluate the accuracy of the model developed using the proposed method in comparison with the classical approach, a synthetic prediction problem was selected. This problem involved forecasting values from an artificial dataset designed to meet the following criteria:

- The input consists of a single numerical value.
- For each input in the training set, there are two distinct correct output values.

Given these conditions, a training dataset was generated using the following algorithm: 1,000 input values were uniformly sampled from the interval $[-15,15]$. For each generated input value x, two corresponding output values were calculated using formulas (7,8):

$$y_1 = \sin(\phi_0 + \phi_1 \cdot x) \cdot e^{-\frac{(\phi_0+\phi_1 \cdot x)^2}{32}} + 0.01 \cdot \varepsilon_1, \tag{7}$$

$$y_2 = \log(x + 15) + 0.01 \cdot \varepsilon_2 \tag{8}$$

In this experiment, the parameters ϕ_0 and ϕ_1 were set to 3 and 0.9, respectively. The noise terms ε_1 and ε_2 were randomly sampled from a normal distribution to introduce variability into the data. As a result, the dataset was obtained, visualization of which can be observed on Figure 1.

In this experiment, the classical MDN model was compared with the previously proposed modified version. The number of Gaussian components in the mixture was set to 2. Both neural network models share the same architecture: a shallow neural network with a single hidden layer consisting of 20 hidden units. The tanh function was selected as the activation function. The full-batch Adam optimizer was used as the model fitting strategy.

For the modified model, the hyperparameter ε was set to 0.05, and the step size for computing the definite integral was set to 0.0025. Both models were initialized such that their trainable parameters had identical values before training. Therefore, both models had the same initial accuracy.

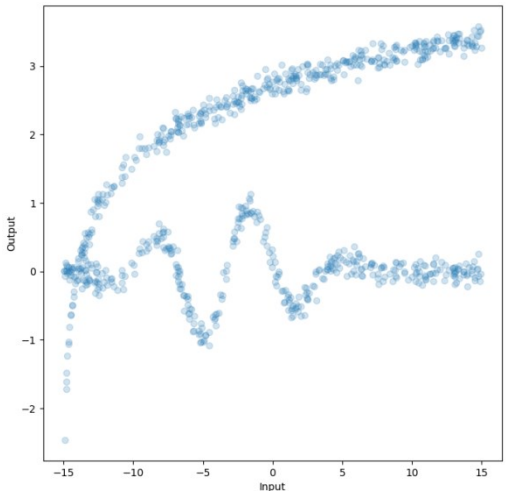

Fig. 1. Generated data visualization

The generated dataset was split into training (80% of the original data) and test (20%) sets. Both models were trained for 1500 epochs. The results of the experiment are presented in Table 1.

Table 1

Comparison of models learning effectiveness on mandatory generated data with multimodal distribution

	Final training loss	**Final testing loss**	**Accumulative training loss (from epoch 500)**	**Accumulative testing loss (from epoch 500)**
Original MDN model	1.208	1.173	1337.074	1318.767
Modified MDN model	1.152	1.120	1304.965	1291.521
Modified MDN model accuracy gain (%)	4.559	4.561	2.401	2.066

The comparison of learning dynamic for both models can be observed on Figure 2.

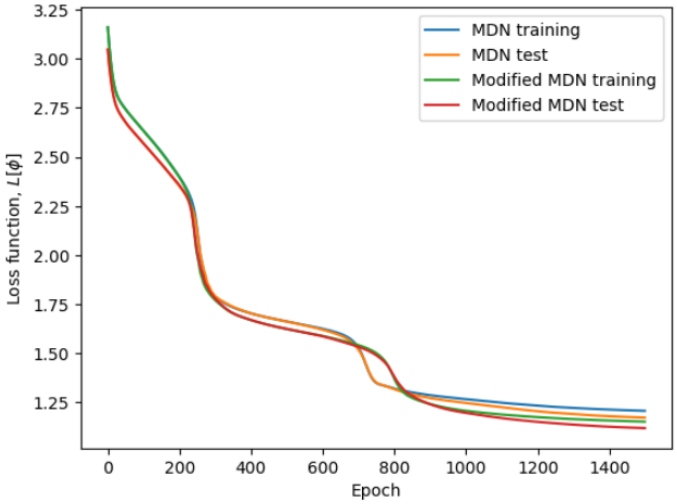

Fig. 2. Learning dynamic comparison for original and modified MDN models in terms of improving a loss values for training and test data

Next, we will compare the prediction accuracy of the models in more practical scenarios. As an example of applying a predictive model to data with a multimodal distribution, we consider a problem related to the kinematics of a robotic arm.

Let us describe this problem in more detail. Imagine a robotic arm consisting of two segments connected together and operating within a single plane. One end of the first segment is fixed in space, while the other end is connected to one end of the second segment. Both segments can rotate within certain limits. The task is to predict the angles of rotation for each segment, given the coordinates of a target point, so that the free end of the arm reaches as close as possible to that point.

From a mathematical perspective, the coordinates x_1 and x_2 of the free end of the robotic arm, with segment lengths L_1 and L_2, and rotation angles θ_1 and θ_2, are given by the following formulas (9, 10):

$$x_1 = L_1 \cos(\theta_1) - L_2 \cos(\theta_1 + \theta_2), \quad (9)$$

$$x_2 = L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) \quad (10)$$

A visual representation of this setup can be seen in Figure 3.

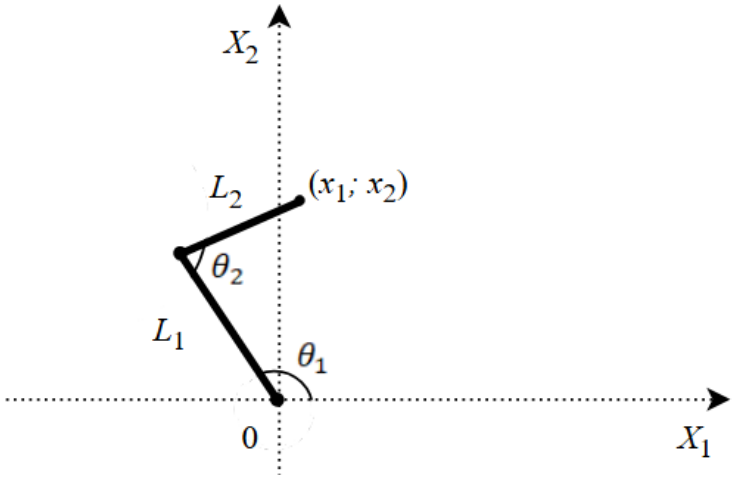

Fig. 3. A schematic representation of a 2-dimensional robotic arm consisting of two segments with lengths L_1 and L_2. When rotated by angles θ_1 and θ_2, respectively, according to the aforementioned formulas, the free end reaches a point with coordinates (x_1, x_2).

This problem involves predicting data with a multimodal distribution, as for certain coordinates, there may exist two different sets of angles that position the robotic arm's end at the same target point. In this study, an experiment was conducted to compare the prediction accuracy for this task using the original and the modified MDN models, as well as a neural network model trained using the least squares method. As the accuracy metric for model prediction, distance error was chosen. This metric measures the average distance between the coordinates reached by the tip of the robotic arm, using the predicted angles, and the expected coordinates.

For the purpose of this experiment, a dataset consisting of 1,000 input-output pairs was generated. The following parameters were defined for the simulated robotic arm:

- $L_1 = 0.815$
- $L_2 = 0.475$
- Minimum rotation angle $\theta_1 = \frac{\pi}{8}$
- Maximum rotation angle $\theta_1 = \frac{5\pi}{8}$
- Minimum rotation angle $\theta_2 = \frac{\pi}{2}$
- Maximum rotation angle $\theta_2 = \frac{3\pi}{2}$

As a result, a dataset was obtained, the visualization of which can be seen in Figure 4.

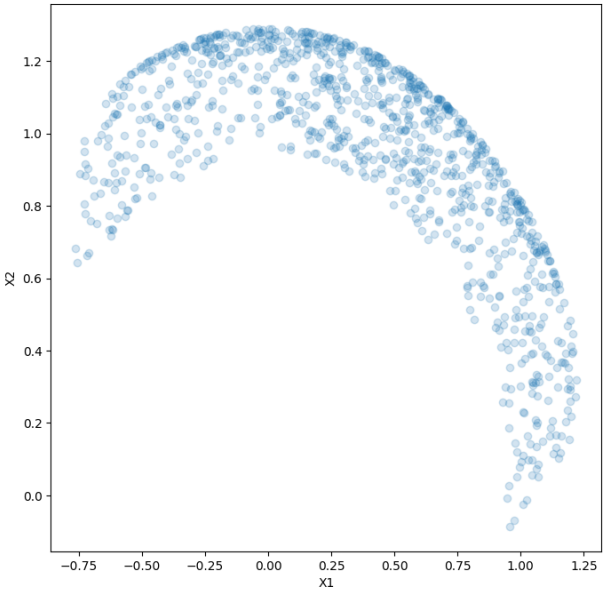

Fig. 4. Visualization of generated coordinates of target points to be reached by the tip of the robotic arm

The following architectural settings were selected for the models tested in this task:

- The MDN-based models use 2 Gaussian components.
- All models consist of an input layer with 2 inputs and a hidden layer with 20 hidden units.

- The MDN-based models have an output layer with 12 outputs.
- The model trained using the least squares method has an output layer with 2 outputs.
- Activation function: Tanh.
- Training strategy: full-batch Adam optimizer.

For the modified model, the hyperparameter ε was set to 0.2, and the step size for calculating the definite integral was set to 0.01. The generated dataset was split into a training set (80% of the original data) and a test set (20%). All models were trained for 1000 epochs. The results of the experiment are presented in Table 2.

Table 2

Comparison of models predicting accuracy for 2-dimensional robotic arm movement

	Distance error	**Accumulative distance error (from epoch 200)**
Least square model	0.154	182.645
Original MDN model	0.121	115.610
Modified MDN model	0.107	107.909
Modified MDN model gain compared to least square (%)	30.584	40.919
Modified MDN model gain compared to original MDN (%)	11.573	6.661

The comparison of dynamic of distance error values for all models during learning epochs can be observed on Figure 5.

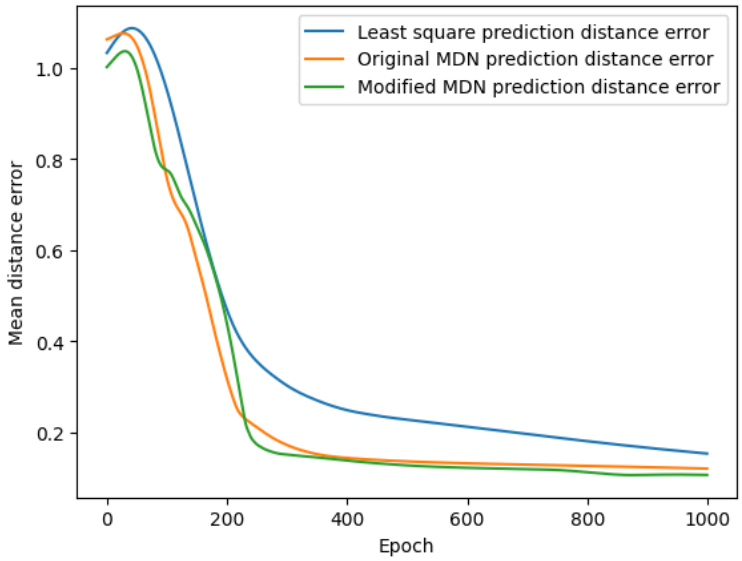

Fig. 5. Distance error values dynamic for all models

Finally, it was decided to complicate the last task and modify it as follows:

- The prediction is performed for two robotic arms simultaneously.
- The reference point of the second arm is shifted along the X_2 axis by a known value called *offset*.
- The angles applied to the second arm rotate it in the opposite direction relative to the first one.
- Both arms have identical length parameters and joint angle constraints.

Accordingly, we obtain the following formulas (11, 12) for the coordinates x_3 and x_4 of the end-effector of the second arm when its segments of lengths L_1 and L_2 are rotated by angles θ_3 and θ_4:

$$x_3 = L_1 \cdot \cos(-\theta_3) - L_2 \cdot \cos(-\theta_3 - \theta_4), \quad (11)$$

$$x_4 = L_1 \cdot \sin(-\theta_3) - L_2 \cdot \sin(-\theta_3 - \theta_4) - offset \quad (12)$$

The parameters were set to be the same as in the previous task. For the second arm, the *offset* parameter was set to 1. Based on this, a dataset containing 1000 input-output data instances was generated, with its visualization shown in Figure 6.

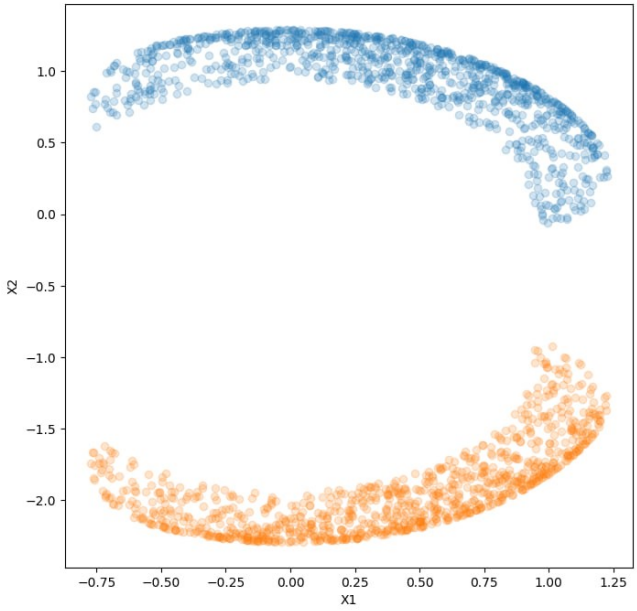

Fig. 6. Visualization of generated coordinates of target points to be reached by the tips of two robotic arms

To solve the prediction task, the same models as in the previous task were selected, with modifications to their input and output layers to match the requirements of the new task. The generated dataset was split into a training set (80% of the total) and a test set (20% of the total). All models were trained for 1000 epochs. The results of the experiment are presented in Table 3.

Table 3

Comparison of models predicting accuracy for two 2-dimensional robotic arms movement

	Distance error	Accumulative distance error (from epoch 200)
Least square model	1.113	944.796
Original MDN model	1.046	847.868
Modified MDN model	1.020	835.355
Modified MDN model gain compared to least square (%)	8.392	11.584
Modified MDN model gain compared to original MDN (%)	2.467	1.476

The comparison of the distance error dynamics for each model during training can be observed in Figure 7.

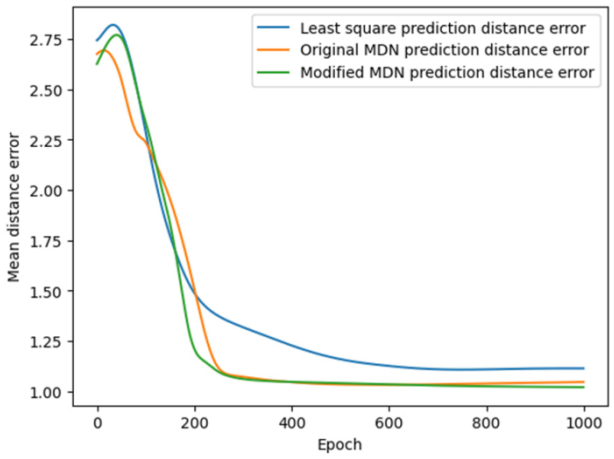

Fig. 7. Distance error values dynamic for all models

Conclusions

This work presented the development and evaluation of an algorithmic and software method for constructing neural network models aimed at predicting data characterized by multimodal distributions. A thorough analysis of existing MDN-based approaches was carried out, followed by the proposal of a modified Mixture Density Network

(MDN) framework. This modification introduces a probabilistic approach into the loss function calculation by employing a numerical approximation of the definite integral, thereby replacing the conventional negative log-likelihood method used in standard MDNs.

The proposed approach was validated through a series of experiments on both synthetic and real-world-inspired datasets. Specifically, predictive tasks involving data with dual-mode output distributions and robotic arm positioning scenarios demonstrated that the modified MDN consistently achieved better learning dynamics and lower prediction errors compared to the classical MDN and traditional least squares models. This performance improvement was evident not only in isolated predictions but also in more complex systems simulating the simultaneous operation of two robotic arms.

Despite these advantages, the enhanced model incurs a significant increase in training time due to the additional computational overhead introduced by the numerical integration step. However, the runtime performance during inference remains unaffected, affirming the practical applicability of the method in real-time systems.

Future research will focus on optimizing the computational cost associated with the modified loss function. Promising directions include exploring alternative analytical or approximate methods for computing the definite integral in the Gaussian mixture context, and investigating surrogate loss functions that retain the model's interpretability and stability while reducing training time. Such advancements could further improve the scalability of this approach in complex multimodal systems, enhancing its value for applications in robotics, image analysis, and beyond.

References

1. Santos, M. S., Abreu, P. H., Japkowicz, N., Fernández, A., & Santos, J. (2023). A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research. *Information Fusion*, *89*, 228-253.

2. Abkar, A. A., Sharifi, M. A., & Mulder, N. J. (2000). Likelihood-based image segmentation and classification: a framework for the integration of expert knowledge in image classification procedures. *International Journal of Applied Earth Observation and Geoinformation*, *2*(2), 104-119.

3. Bishop, C. M. (1994). Mixture density networks. *Technical Report NCRG/94/004, Neural Computing Research Group, Aston University*.

4. Liu, P. L. (2019, September 30). *Multimodal regression — Beyond L1 and L2 loss*. Medium. https://medium.com/data-science/anchors-and-multi-bin-loss-for-multi-modal-target-regression-647ea1974617

5. Suzuki, M., & Matsuo, Y. (2022). A survey of multimodal deep generative models. *arXiv preprint arXiv:2207.02127*. https://arxiv.org/abs/2207.02127

6. Li, Y., Zhang, Z., & Wang, Y. (2022). Physics-guided mixture density networks for uncertainty quantification in structural reliability analysis. *Reliability Engineering & System Safety, 222*, 108443.

7. Campbell, J., Stepputtis, S., & Ben Amor, H. (2019). Probabilistic multimodal modeling for human-robot interaction tasks. *Proceedings of Robotics: Science and Systems (RSS)*. https://doi.org/10.15607/RSS.2019.XV.047

8. Rahimi, A., & Alahi, A. (2024). A multi-loss strategy for vehicle trajectory prediction: Combining off-road, diversity, and directional consistency losses. *arXiv preprint arXiv:2411.19747*. https://arxiv.org/abs/2411.19747

9. Qian, Y., Guo, Z., Xie, H., Zhou, C., & Chen, H. (2024). Multimodal protein function prediction with integrated sequence and structure representations. *Scientific Reports, 14*(1), 1843.