COMP5212 Project

Zijian Zhao

21071818

1 Methodology

Overview

In this project, we implement a deep learning model for regression of agricultural equipment sale prices. We observe that the data presents several significant challenges that complicate model training:

- **High Noise:** Occasionally, two similar samples exhibit significantly different sale prices. This discrepancy arises because the sale price depends not only on the given attributes, such as equipment type or year, but also on human factors.
- **Mixed-Type Input Data:** The input data comprises discrete features, including manufacturer, model, gearbox type, and fuel type, as well as continuous features such as year, operating hours, efficiency, and engine capacity.
- **Missing Types:** For the attribute of model, we notice that some models appear in the testing set but are absent from the training set.
- **High Variability in Ground Truth:** We observe that the prices in the training set exhibit a wide range, varying from 795 to 126,000. This high variability can negatively impact model training.

To address these challenges, we construct a deep learning model with two main design strategies:

- **Mitigating High Noise:** Following the approach outlined in (Piovesan et al., 2023), we utilize a Gaussian distribution to predict prices rather than focusing on a specific value, thereby reducing the influence of noise. This approach allows the variance component of the Gaussian distribution to help the model capture the uncertainty in the predictions. Additionally, we adopt a method from our previous work (Chen et al., 2024) that employs separate upstream bottom layers to process continuous and discrete features independently before fusing them into a shared upstream head layer.
- **Mask Learning Method:** In our earlier work (Chen et al., 2024), we encountered a similar issue with missing types in base station modeling. We proposed a mask learning method that randomly masks input data during training with a [MASK] token, which is then used to replace the missing type in the testing set during inference.

In addition to these strategies, we also conduct thorough data preprocessing to enhance model performance. First, we observe that there is only one sample with a fuel type of 'Electric'. To mitigate its influence, we simply remove it from the training set. Next, to address the high variability in the ground truth, we calculate the mean and standard deviation of the training set ground truth, which are 16849.08 and 9846.69, respectively. During model training, we normalize the ground truth using the formula $\frac{y-16849.08}{9846.69}$. Subsequently, during inference, we re-normalize the predicted values using the equation $9846.69\hat{y} + 16849.08$.

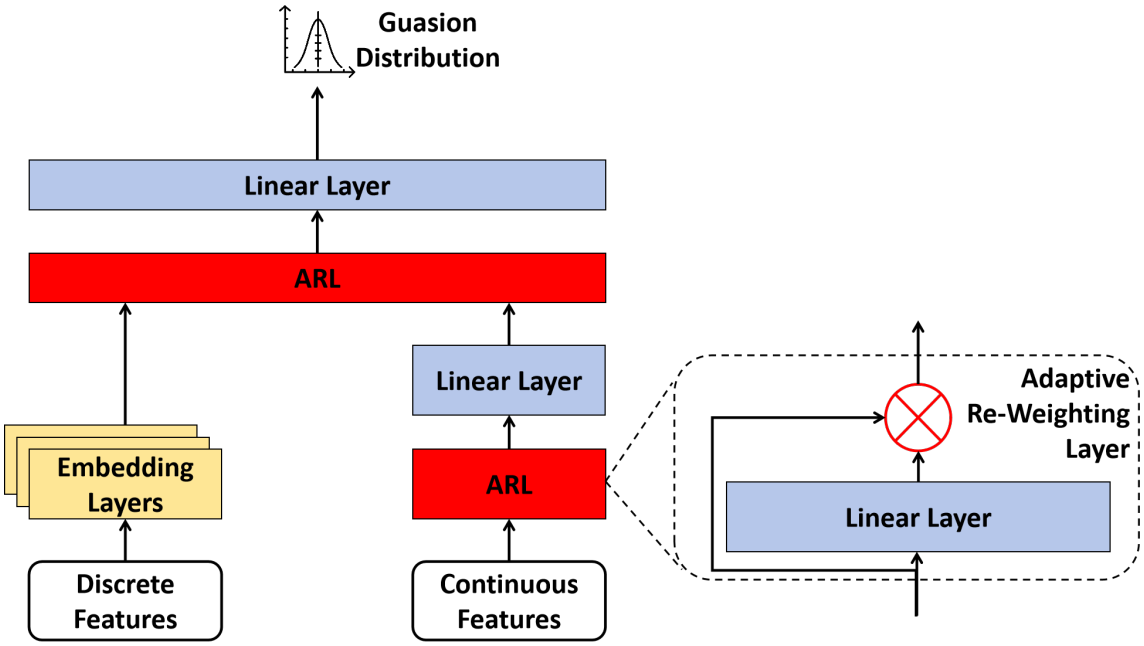

Figure 1: Model Architecture

Model Structure

Our model structure is illustrated in Fig. 1. First, we extract features from the discrete and continuous inputs separately. For discrete inputs, we use a separate embedding layer for each attribute. For continuous inputs, we begin with an Adaptive Re-Weighting Layer (ARL) (Chen et al., 2024) to assign different weights to different attributes. The intuition behind ARL is similar to that of LassoNet (Lemhadri et al., 2021), but it is implemented in a simpler manner:

$$\begin{aligned} w &= \text{LinearLayer}(x) \ , \\ y &= x \circ w \ , \end{aligned} \tag{1}$$

where w represents the weight of each attribute and $\circ$ denotes element-wise multiplication. In this way, we aim for the model to assign higher weights to the more important attributes. Following this, we use a Linear Layer to extract features from these continuous inputs. Finally, we concatenate the feature embeddings from all discrete and continuous features and input them into another ARL and Linear Layer. After the last layer, the model generates the estimated price as a Gaussian distribution $N(\mu, \sigma)$ by outputting the two parameters.

Training Method

Since the model outputs a Gaussian distribution rather than a single value, we cannot directly use traditional loss functions like Mean Squared Error (MSE) or Mean Absolute Percentage Error (MAPE). Instead, following the principle of Maximum Likelihood Estimation (MLE), we define the following loss function:

$$\begin{aligned} \theta &= \arg\max_{\theta} P(y|N(\mu(x;\theta), \sigma(x;\theta)^2)) \\ l(\mu, \sigma, y) &= -\log P(y|N(\mu, \sigma^2)) \\ &= \log(\sigma) + \frac{(y-\mu)^2}{\sigma} \ , \end{aligned} \tag{2}$$

where θ represents the model parameters, x is the model input, μ and σ are the model outputs, and y is the ground truth. This approach aims to ensure that y has a high

Table 1: Ablation Study

Model	Proposed	w/o Gaussian	w/o ARL	w/o Gaussian & ARL
RMSE	**2209.6**	2683.7	2426.5	2868.7

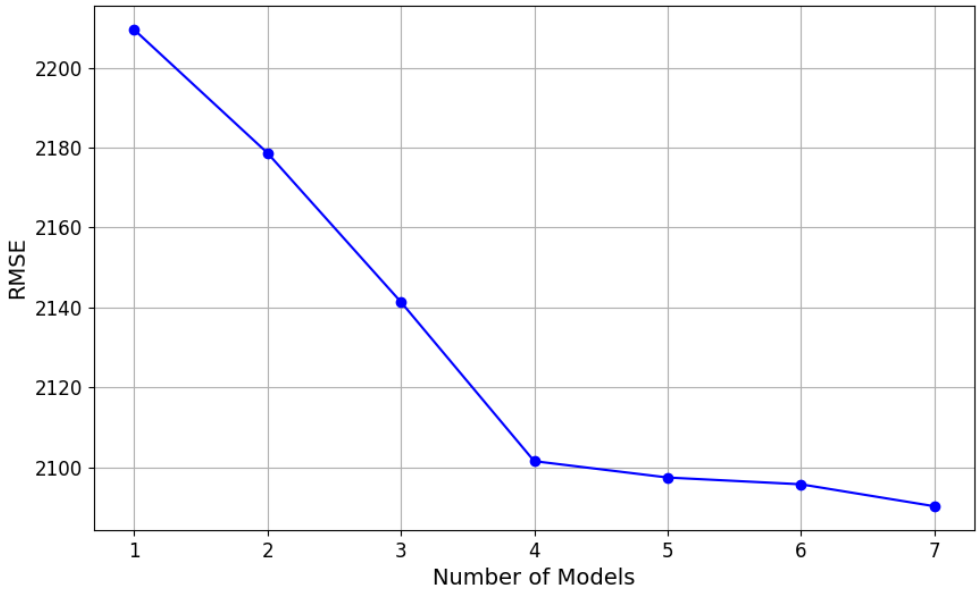

Figure 2: Model Performance with Varying Numbers of Models

probability within the predicted Gaussian distribution.

To address the issue of certain model types in the testing set not being present in the training set, we design a MASK-learning method. In each training epoch, we randomly replace the model of 15% of the samples in the training set with a [MASK] token. In the testing set, we similarly use [MASK] to replace those models that are not in the training set.

Finally, to prevent overfitting, we randomly allocate 10% of the training set as a validation set. If we observe that the loss on the validation set does not decrease for 30 consecutive epochs, we stop the model training. However, because many attributes only appear in a single sample within the training set, this random split may result in some important features being excluded from the training data. To mitigate this issue, we employ an ensembling method. We train multiple models using different splits of the training and validation sets. During inference, we take the average of the predicted values μ as the final result.

2 Experiment Result

To illustrate our model's performance, we initially set aside the ensembling component and train a single model for comparison. We conduct a series of ablation studies to demonstrate the effectiveness of our ARL module and the Gaussian distribution module. For the sake of fairness, we report the model's performance on the public testing set. The results are presented in Table 1.

For the ensembling part, we aim to determine the optimal number of models. The experiment is illustrated in Fig. 2. It can be observed that the model performance increases at a slower rate after the number of models reaches 4.

References

Chen, T., Wang, Y., Chen, H., Zhao, Z., Li, X., Piovesan, N., ... Shi, Q. (2024). Modelling the 5g energy consumption using real-world data: Energy fingerprint is all you need. *arXiv preprint arXiv:2406.16929*.

Lemhadri, I., Ruan, F., Abraham, L., & Tibshirani, R. (2021). Lassonet: A neural network with feature sparsity. *Journal of Machine Learning Research*, *22*(127), 1–29.

Piovesan, N., López-Pérez, D., De Domenico, A., Geng, X., & Bao, H. (2023). Power consumption modeling of 5g multi-carrier base stations: A machine learning approach. In *Icc 2023-ieee international conference on communications* (pp. 3633–3638).