Homework data science lab: Regression task on particle position estimation

Mattia Ottoborgo
Federico Fortunati
Politecnico di Torino
Student id: s323001 s328854
s323001@studenti.polito.it
s328854@studenti.polito.it

***Abstract*—In this report we propose a possible approach to the regression task applied to the particle position estimation. It consists in the removal of noise components to identify the features related to the 12 pads. These features are then preprocessed and fed into a Random Forest and a Multi-layer Perceptron regressors. Among the two models, the MLP yields a lower MSE. Both proposed solution outperform the naive baseline defined for the problem.**

I. Problem overview

The proposed project is a regression problem on a Dataset containing information about the passage of a particle through a sensor. A sensor is composed by 12 pads, and every pad detects the passage of a particle, called event, and collects the following characteristics: pmax, the magnitude of the positive peak of the signal, in mV; negpmax the magnitude of the negative peak of the signal, in mV; tmax, the delay (in ns) from a reference time when the positive peak of the signal occurs; area,the area under the signal; rms, the delay (in ns) from a reference time when the positive peak of the signal occurs. The goal of the project is to correctly predict, for each event, the coordinates (x,y) where the particle of interest passed. The dataset is divided into two parts:

- a *development* set, containing 385,500 events
- a evaluation set, comprised of 128,000 events.

In the dataset we have 18 readings, whereas the number of pads is 12. This occurs because of hardware constraints in the data acquisition phase, and a subset of the features contains noise.

Figure 1 shows the distribution of pmax between the 18 features of the first 500 readings. Upon close examination, several features exhibits notable variations in magnitude compared to the others. Specifically, features indexed as pmax[15],pmax[16] and pmax[17] stand out due to their higher values. This observed disparity may be attributed to the noise disturbances, as previously mentioned.

Figure 2 shows a graphical representation that captures the positions of pmax values for each reading, where a color-coded scheme is employed to highlight areas with varying frequencies of pmax occurrences. We notice some uniformly clear or dark regions, such as those at positions 0, 7, 12, 16, and 17, and therefore we consider them potentially uninteresting and indicative of noise.

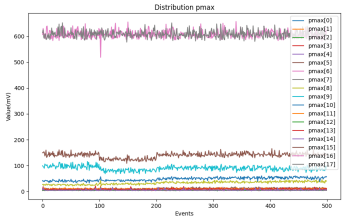

Fig. 1. Distribution of pmax

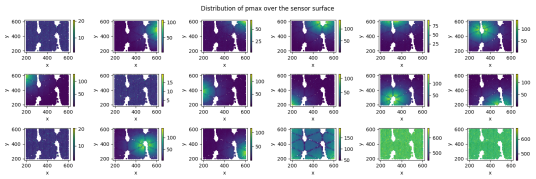

Fig. 2. Distribution of x and y

Finally in Figure 3 it is depicted an estimation of sensor pads according to the measures in the training set. The approximation for the i-th feature is calculated as follows:

1) Retrieve the highest 8000 samples of the feature pmax[i].
2) Calculate the average x and average y.
3) Repeat for all pmax features.

Based on the sensor description proposed in [1] and the visualization in Figure 4, we managed to isolate twelve features as sensors.

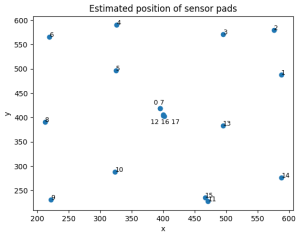

Fig. 3. Estimated position of sensors pads

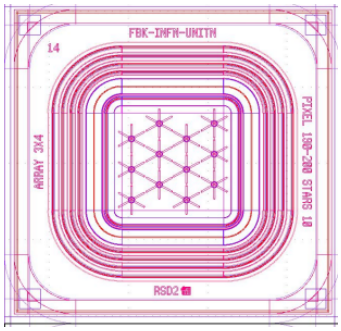

Fig. 4. RSD sensor structure

II. Proposed approach

Our approach can be described according to these three phases:

A. Preprocessing

All records do not present null or corrupted data, hence we did not proceed with data cleaning. We have then evaluated different approaches:

- log transformation to reduce the effect of outliers.
- No transformation applied.
- Normalise each feature to the total of the same subgroup. Let us consider for example the subgroup of n features pmax. Given the i-th feature of this subgroup, we have calculated "pmax[i]_norm" such that:

$$\texttt{pmax[i]\_norm}=\frac{pmax[i]}{\sum_{i=1}^{n} pmax[i]}$$

By executing different tests, we observed that the model trained on preprocessed according to the second option described yields the best performances.

B. Model selection

The model we have tested have been:

- Linear Regression: this is a supervised learning algorithm used for predicting a continuous outcome variable based on one or more predictor variables. The model assumes a linear relationship between the predictors and the target variable.
- Ridge: this is a regularization technique used in linear regression to prevent overfitting by adding a penalty term based on the sum of squared coefficients to the traditional least squares objective.
- Lasso: this is another regularization method in linear regression, aiming to prevent overfitting by adding a penalty term based on the absolute values of the coefficients.
- Decision Tree: this algorithm recursively splits the dataset into subsets based on the most significant feature at each node, creating a tree-like structure of decisions.
- Random Forest [2]: this algorithm builds multiple decision trees, each on a random subset of the data with bootstrapping. This typically avoids the overfitting problem of decision trees but still maintaining interpretability.
- Gradient Boosting. This algorithm builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function.
- Multi-layer Perceptron regressor. A feedforward artificial neural network, consisting of fully connected neurons with a nonlinear kind of activation function, organized in at least three layers.

We evaluated the models with Cross validation on the MSE score. From Table 1, the Random Forest achieves the best performances with the highest MSE. Hence, we decide to proceed with both the Random Forest and the MLP, since it may perform better with the right parameters.

model	MSE	R2
Random Forest	15.64	0.9988
Decision Tree	35.26	0.997
Ridge	108.44	0.992
Linear	106.89	0.992
Lasso	244.75	0.982
Gradient Boosting	17.59	0.998
Multi-layer Perceptron	17.23	0.997

TABLE I
TABLE 1 : RESULTS OF MODEL SELECTION PROCESS.

C. Hyperparameters tuning

After choosing the model to continue our regression task, we decided to experiment with different parameters during the training phase. In particular we have tuned:
Random Forest

- "n_estimators", which indicates the number of tree composing the Random forest model;
- "criterion", which determines the function used to evaluate the quality of a split.
- "max_features", which determines the number of features to consider when looking for the best split.
- "cv", which determines the cross-validation splitting strategy.

MLP

- ”activation” ,which determined the activation function used by the algorithm.
- "hidden_layer_sizes" , which determines the number of neurons in the hidden layer.
- "learning_rate" , which determines the learning rate schedule for weight updates.

III. Results

We decided to choose the model according to the best MSE score, since this is the metric indicated in the project description. Based on this, we have decided to proceed with the hyperparameter tuning on the Random Forest Model and the MLP. This step yielded the following parameters through k-fold cross-validation:

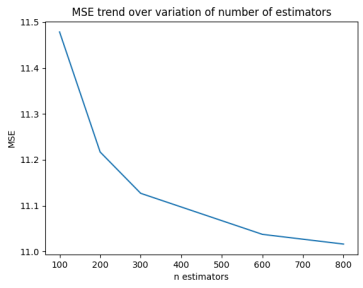

Fig. 5. MSE trend over n estimators of random forest

Random Forest

- `"n_estimators"` : 800;
- `"criterion"` : `"squared_error"`;
- `"max_features"` : `"sqrt"`;

Multi-layer Perceptron

- ”activation” : ”logistic”
- `"hidden_layer_sizes"` : 8000
- `"learning_rate"` : ”adaptive”

From Figure 5, it is possible to analyse the trend of the MSE score with the increase of the number of estimators for the random forest.

The Random forest and the MLP with the above mentioned characteristics yielded respectively an MSE of 11.01 and 16.09 in the development dataset. However, the public scores for these models were 4.614 and 4.465, respectively. The result is interesting because, based on the MSE got on the development dataset, we expected that the random forest would have outperformed the Multi-layer Perceptron, but the opposite occurred. A possible reason for this counterintuitive result may be the overfitting of the random forest model, whose performance degrades in the public score because it loses generality. We may reasonably assume that the private score will return better performance for MLP with respect to the Random Forest.

For comparison, a naive solution has also been defined. The approach involves utilizing all data without any preprocessing: it simply applies a random forest model with 10 estimators. This solution obtains a public score of 6.026, closely approaching the baseline.

IV. Discussion

The proposed approach yields results that outperform the naive baseline defined. The following are some aspects that might be worth considering to further improve the obtained results:

- Better handling of data noise. Currently no preprocessing operation aim to filter outliers or remove noise, as we do not possess enough domain knowledge to discriminate noise from particular events.
- Our approach utilised three different analysis to identify the six sources of noises. However, a further investigation is necessary to make sure that the selection we have proposed is correct.
- During the analysis phase, we have noticed that some features within the pmax group present some similarities. In particular they present an interesting pattern similar to a latency. This pattern can be easily represented by Figure 6. This is due to the fact that after the particle impact, the signal require a certain amount of time to propagate to all the pads. It would be interesting to try to take into account the above mentioned latency as feature
- The MLP is outperforming the random forest even though its MSE is higher. We suppose this is due to a possible overfitting of the Random forest which makes the model to perform worse in the public score.
- The úsage of MLP yielded the best score for us in the public score. More advanced Neural Network could be investigated to further improve performances.
- An interesting extension of this research could be the implementation of a data augmentation technique as described in [3]. This would allow the model to perform better in areas poorly represented in the dataset and in general enhancing the overall performance by providing data that contains events with multiple amplitudes.

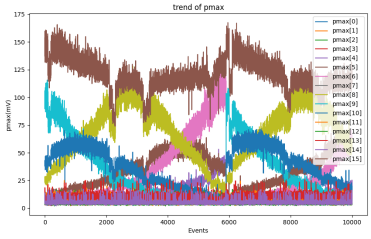

Fig. 6. Trend of pmax in different measurements

References

[1] M.Mandurrino, *The second production of RSD (AC-LGAD) at FBK,*. JINST, 2022.
[2] L. Breiman, *Random forests*.
[3] F.Siviero, *First application of machine learning algorithms to the position reconstruction in Resistive Silicon Detectors*. JINST, 2021.