Machine Learning Project

Quantum Information and processing

Aristotle University of Thessaloniki
Physics Department
Msc Computational Physics
Bompotas Christos
AEM:4435
20/06/2024

Contents

I Abstract

In this project, we utilized two different classifiers, along with an artificial neural network (ANN) classifier, to predict whether events recorded in the Large Hadron Collider and the Tevatron Colliders are background noise or useful signals. Using the provided CSV file, we divided the data into low-level and high-level features. The input signal classified each event as either background noise or a signal.

For each case we divide the data into high and low level features and set the target variable, split the data into test and train data and scale them. We chose the train to test data ratio to be 9/1. Note that inside the csv file was a string value which was changed manually inside the csv. Of course for many cases of such errors a computational solution would be necessary that checks for string values and replaces them with floats.

1 K-Nearest-Neighbor

We start with the k-nearest-neighbor classifier, chosen for its simple implementation and its versatility in handling various types of data. Through trial and error we concluded that the best possible accuracy was achieved for $k = 10$ for the low level features and $k = 9$ for the high level feautures. The classifier performance is evalueated for each level using confusion matrices, accuracy scores, along with an area under the curve (AUC) score and the corresponding ROC curves.The confusion matrices highlight the number of true positives, true negatives, false positives, and false negatives, offering insight into the classifier's precision and recall. For the low-level features, the confusion matrix results were as follows: 226 true positives, 249 true negatives, and 163 false positives and negatives. For the high-level features, the confusion matrix showed 245 true positives, 299 true negatives, 144 false positives, and 113 false negatives The ROC curves and the corresponding AUC scores indicate the model's ability to distinguish between the positive and negative classes across all thresholds. The accuracy scores were 0.579 or 57.9% for the low level features and 0.671 or 67.1% for the high level. The AUC scores were 0.589 or 58.9% for the low level and 0.66 or 66% for the high level. The AUC score is more trustworthy than accuracy, because of its robustness in class imbalance and shows the probability of classifying right a random piece of data.

2 Support-Vector-Machine

The second classifier we used was a a Support Vector Machine (SVM) classifier, known for its versatility by using different kernel functions. For our case we utilised a linear kernel, which is the simplest form, and particularly suited for datasets where the classes are linearly separable or nearly so. Again the classifier was applied to both levels and its performance was assessed using confusion matrices and accuracy scores, together with ROC curves and AUC scores. For the low-level features, the confusion matrix indicated 157 true positives, 287 true negatives, 232 false positives, and 125 false negatives. The fact that the false positives number is bigger than the true positives sould be an indicator to experiment with the other kernels for more trustworthy results. For the high-level features, the confusion matrix showed 65 true negatives, 60 true positives, 15 false positives, and 25 false negatives. The accuracy scores were 0.554 or 55.4% for the low level features and 0.629 or 62.9% for the high level features. The AUC scores were 0.59 or 59% and 0.66 or 66% for the low and high level respectively.

3 Artificial-Neural-Network

In this last section we used a neural network classifier using PyTorch. We define a single network for each level using a feedforward architecture. Each network consists of two hidden layers with 12 and 8 neurons, respectively, each followed by a ReLU activation function, and a single neuron output layer with a Sigmoid activation function for binary classification. The networks are trained separately for low-level and high-level features using the Adam optimizer and Binary Cross-Entropy Loss (BCELoss). The training process involves loading the data into DataLoader to enable batch processing. During training, the optimizer gradients are reset at the beginning of each batch. The forward pass computes predictions, the loss is calculated, and backpropagation is performed to update weights. This process is repeated for multiple epochs to optimize the network parameters.

After training, the networks are evaluated on the test set. Predictions are generated in a no-grad context to prevent gradient computation. Test loss and accuracy are calculated to assess performance. For a detailed evaluation, confusion matrices and ROC curves were plotted again. For low-level features, the test loss was 68.9% , with a test accuracy of 59.2%. The confusion matrix indicated that out of the test samples, 214 true positives and 260 true negatives were correctly classified, while 175 false positives and 152 false negatives were misclassified.The ROC AUC score for low-level features was 63.5%, indicating a satisfactory true positive rate with a low false positive rate. For high-level features, the test loss is 56.8%, with a test accuracy of 70.7%. The confusion matrix showed 261 true positives and 305 true negatives correctly classified, with 128 false positives and 107 false negatives misclassified. The ROC AUC score for high-level features is 77.5%, suggesting correct discriminative ability.

4 Conclusions

All in all, KNN is the simplest method that classifies a sample based on the majority class of its k-nearest neighbors in the feature space. It is simple, intuitive, and effective for datasets with well-defined boundaries. SVM, on the other hand, finds the hyperplane that maximizes the margin between two classes in the feature space and can handle non-linear data using kernel functions. Neural networks, in contrast, learn hierarchical representations of data through multiple layers of interconnected neurons, allowing them to model complex relationships and patterns in data. The results demonstrate that neural networks can achieve high accuracy and AUC scores, making them suitable for reliable results, far greater than that of the typical classifiers. It is obvious nonetheless that these results come at the cost of increased computational requirements and the need for larger datasets.