Classifying Daily Activities Based on Inertial Sensor Data

Miguel Barrios (CS 4033) and Justin Santos (CS 5033)
The University of Oklahoma

Abstract

Classifying a person's activity based on sensor measurements can prove highly important. In fact, various mobile devices already have software that is able to do so with high accuracy. In this project, various machine learning algorithms and techniques are tested to determine someone's current activity based on inertial sensor measurements.

1. Introduction

The domain of this project is human activity recognition, a field-centered around the identification of human activity. The identification is achieved by utilizing machine learning algorithms on data collected from various devices such as accelerometers, motion sensors, and cameras. The dataset used for this project comes from the UCI Machine Learning Repository. Specifically, the dataset is entitled "Daily and Sports Activities Data Set."

In this dataset, 19 varying activities are carried out for a total of 5 minutes by 4 male subjects and 4 female subjects. The exhaustive list of each activity is listed in Table 1 of the appendix. The subjects were instructed to perform the activities in his or her own personal style and in an unrestricted manner. Each subject is fitted with an inertial sensor on their torso, left arm, right arm, left leg, and right leg. Data is recorded in 5-second intervals, which includes 3-dimensional data from gyroscopes, accelerometers, and magnetometers integrated inside of each inertial sensor.

Five experiments were performed on this dataset each aimed at using the inertial sensor data to classify which of the 19 activities the subject is performing.

2. Literature Review

Our first reference by Altun, Barshan, and Tunçel [1] performed many of the same experiments we performed, each aimed at trying to recognize human activity by the measurements provided by body-worn sensors. After acquiring the discrete-time sequence data from the sensors, 6 features were extracted consisting of the minimum and maximum of the following statistics. mean, variance, kurtosis, autocorrelation, and the peaks of the discrete Fourier transformation. The resulting feature extraction produced a total of 1170 features which the group reduced to 30 using principal component analysis. The main contributions we took from this paper is feature extraction and the utilisation of principal component analysis to reduce the number of features.

The second reference by Cunningham and Delany [2] discusses the k-nearest neighbors algorithm. In this paper, the Minkowski distance is discussed, which is a generalized version of Manhattan and Euclidean distances. In fact, Manhattan distance is the 1-norm Minkowski distance, and Euclidean distance is the 2-norm Minkowski distance. Dimensionality reduction techniques such as principal component analysis were also discussed in combination. Since k-nearest neighbors is a lazy learning technique with no actual training step, the algorithm could take an extremely long time to compute all of the distances between points. Principal component analysis could help with this runtime issue by reducing the amount of distances needed to be calculated. In this project, the Minkowski distance is not implemented, instead the simpler formulas of Manhattan and Euclidean distances themselves are calculated. Furthermore, although applying principal component analysis could provide runtime benefits, it was not used in combination with the k-nearest neighbors implemented in this project because the regular runtime did not pose a significant issue in the first place.

The third reference by Kaghyan and Sarukhanyan [3] further discusses k-nearest neighbors, but this time in the domain of human activity recognition. They conducted experiments using a smartphone fitted with a tri-axial accelerometer and attempted to predict one of five activities being done by the smartphone user: standing, walking, running, sitting, climbing up stairs, or climbing down stairs. They used k-nearest neighbors to do their predictions. Their classifier was able to predict standing and sitting activities with 100% accuracy, and all other activities with near-perfect accuracy. This is extremely similar to what is being done in this project. In fact, all of the activities they are attempting to classify are also being classified in this project. The key difference between their project and this project is the sensor suite being used. This project uses gyroscopes, accelerometers, and magnetometers, whereas Kaghyan and Sarukhanyan's experiment only used accelerometers.

The fourth reference Sánchez and Skeie performed a study on human activity classification in a smart house environment using decision trees. The results of this study showed that their decision tree model was able to predict the correct activity with a total accuracy of 88%. This study's sensor suite uses various sensors embedded in furniture and rooms and is vastly different from the sensor suite in this project. However, it still proves that decision trees are useful in the domain of human activity recognition.

The fifth reference by Martínez-Villaseñor and Ponce [4] goes deeper into the field of Human Activity Recognition (HAR). The main idea gotten from this text is the process outlined for the classification of HAR. Once the data has been acquired from the sensors, the data must be split into fixed time windows or sections. In our case, the data was already split into 5-second windows. The next step listed in the reference is feature extraction, in which the reference provides a table of common time-series statistics worth extracting. Once the features have been extracted there is a high possibility of redundancy in the features, which can be worked around by the utilization of principal component analysis. The last step listed in the reference is the model selection, which is just the selection of an appropriate machine learning model to achieve the required result. The reference also goes into other forms of human activity recognition such as the utilization of computer vision.

The sixth reference by Zaki et. al. performs various experiments on human activity classification using many different classification algorithms, similar to this project. They specifically use principal component analysis and k-nearest neighbors, naive Bayes, gradient boosting, random forest, and logistic regression. Their results showed that logistic regression performed most accurately with 96% and 94% accuracy on two different datasets. Naive Bayes performed the worst with 77% and 74% on their respective datasets.

3. Hypotheses

We believe that using a variety of supervised machine learning algorithms trained on the sports and activities dataset, we will be able to accurately predict with at least 80% accuracy which of 19 activates a person is performing. We bring forth the following hypotheses. A multiple layer neural network will perform better than a single layer neural network. A neural network trained on a reduced set of features will have similar performance with a vast reduction in the amount of training time required.

4. Experiments

Experiment 1

The first experiment was aimed at finding out if a model based on a single layer neural network would perform better than one based on a neural network with multiple layers. To compare both models we first needed to find the configurations that produced the best performing model. Several neural networks with varying parameters were created and tested. The performance of each model was determined by how well the model was able to minimize the mean squared error function, and its prediction accuracy on the never before seen data in the test set. The results from the best performing single layer neural network and the neural network containing multiple layers were then compared against each other.

Experiment 1 Results

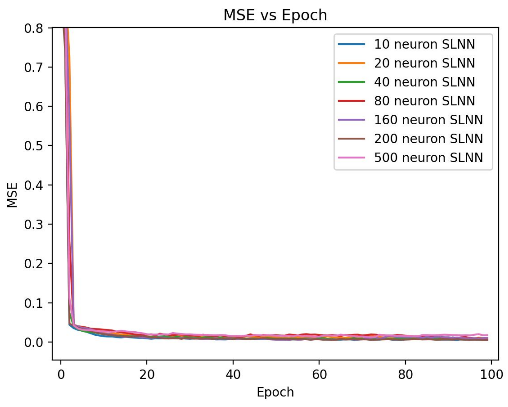

Figure 1: mean squared error progression for each neural net

The results for a single layer neural network's performance in minimizing the mean squared error(MSE) can be seen in the figure above. In the figure we see that every model was able to minimize the MSE to a great extent, however, the neural network containing 200 neurons achieved the lowest average MSE per epoch averaging to just 0.005 after 100 epochs.

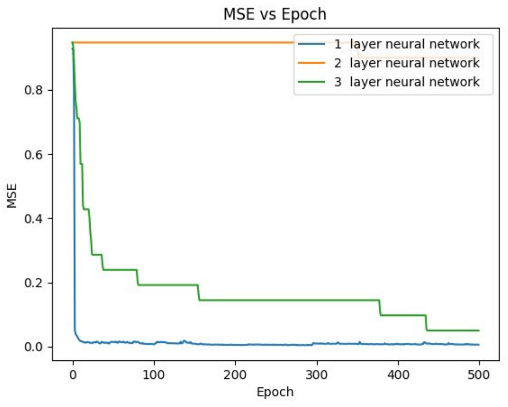

Figure 2: close up mean squared error progression for each neural net

As for the neural network containing multiple layers, the results can be seen in figure 2. This shows that a single layer neural network did a better job at minimizing the mean squared error over the training data. The 3 layers neural network also performed well and might have possibly improved even more if trained for several thousand more iterations. To confirm that the 200 neuron single-layer neural network performed better than the 3 layer neural network both models made predictions on the test set and which were then compared to their actual results. The single-layer neural network correctly classified 1752 of the 1824 examples in the testing set yielding an accuracy of 96 percent. The 3 layer neural network correctly classified 1314 of the 1824 examples yielding an accuracy of 72 percent. This contradicted our original hypothesis that a neural network containing multiple layers would perform better than a single layer neural network. The main reason for this might be the multi layer neural net simply needed substantially more training time to fully converge.

Experiment 2

The second experiment looked into how feature reductions using principal component analysis affects performance. The same methodology was used to find the best parameters for this new neural network as in experiment one. Once the optimal parameters were found a new neural network was trained and tested on the data set with reduced features. The final results were then compared to the results from the best performing neural network in experiment one.

Experiment 2 results

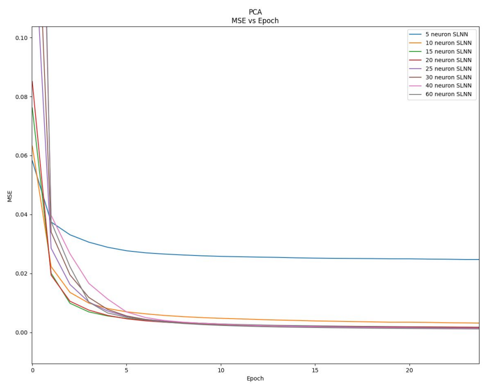

Figure 2.1: mean squared error progression for each neural net trained with reduced features

features	MSE Avg	Epoches	Accuracy
270	0.0020117	2000	0.96
30	0.0005687	50	0.64

Table 1: results of neural net trained with and without PCA

From the figure above, we can see that the best performing model was the single-layer neural network containing 60 neurons. Using that network we made predictions on the test set, which resulted in the correct classification of 1168 of the 1824 training examples, which means the model is able to correctly predict never before seen samples with 64 percent accuracy. This is a 32 percent drop in performance when compared to the neural network trained in experiment one. These results only partially supported the initial hypothesis that a neural network trained on a reduced number of features will have similar performance than one trained on the full set. There are several reasons why this model did not perform as well as the one in experiment one. One possible reason is that many of the classes are very similar to each other and thus cannot be differentiated using the reduced number of features.

Experiment 3

The third experiment used the k-nearest neighbors (k-NN) algorithm in order to classify the type of activity the subject is doing. Hyperparameter searching was done using a grid-search approach. There were two hyperparameters chosen for the search: k-value (1 to 19) and the distance calculation metric (Euclidean, Manhattan, or Chebyshev). Each individual k-NN model in the search is evaluated using prediction accuracy.

This experiment used a training set, validation set, test set, where 60% of the data was used for training, 20% was used for validation, and 20% was used for testing. During the hyperparameter search, the models used were trained on the training set and their accuracies were evaluated using the validation set. After the hyperparameter search, a model was then re-trained using the best hyperparameters on both the training and validation sets. The testing set was then used to evaluate this new model.

Experiment 3 Results

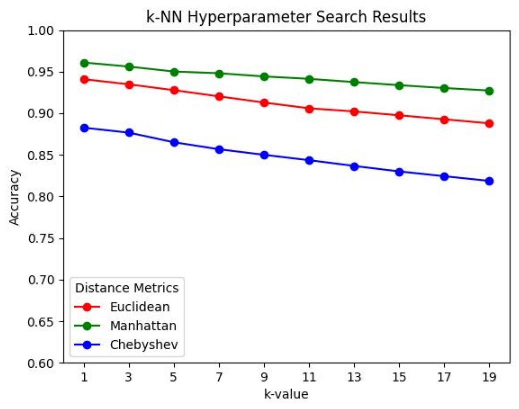

Figure 3: The results of the k-NN hyperparameter search

Figure 3 shows a graph of the results of the hyperparameter search. Manhattan distance is the best distance metric across the board, and Chebyshev distance is by far the worst. The accuracy also consistently decreases as the k-value increases. Interestingly enough, the best models actually come from k = 1, with the best overall model being using a k-value of 1 and Manhattan distance. The hyperparameter search found this model to be 96.1% accurate. When evaluated using the testing set, this model was 95.3% accurate.

Experiment 4

The fourth experiment used decision trees in order to classify the type of activity the subject is doing. Hyperparameter searching was again done using a grid-search approach on two hyperparameters: the maximum depth of the tree (5 to 40) and the criterion to measure the quality of a split (entropy or Gini impurity).
Each individual decision tree in the search is also evaluated using prediction accuracy.

This experiment uses the same training, validation, and testing set split as described in Experiment 3.

Experiment 4 Results

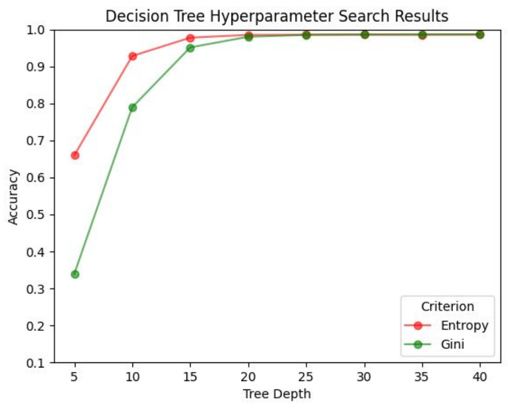

Figure 4: The results of the decision tree hyperparameter search

Figure 4 shows a graph of the results of the decision tree hyperparameter search. Both Gini impurity and entropy eventually converge to the same accuracy, approximately 98%, at depth 25. However, Gini impurity starts off with an accuracy that is significantly worse than entropy. Accuracy increases as the depth of the tree increases, up to a depth of 25; at that point, the accuracy stabilizes. The final model chosen was a decision tree using entropy with a depth of 25, which performed at 98.6% accuracy when evaluated using the validation set. The same model re-trained using both the training and validation sets and evaluated using the testing set had 98.1% accuracy.

Experiment 5

The fifth experiment uses logistic regression in order to classify the type of activity the subject is doing. Since this is a multi-class problem, binary logistic regression will not suffice. Instead, the one-versus-all algorithm (OvA) will be applied, which uses multiple logistic regression classifiers in order to make a multi-class prediction. Since there are 19 classes, there are 19 separate logistic regression classifiers This implementation of OvA is also Justin's novel feature for this project.

To search for the best OvA classifier, six different learning rates were tested and their corresponding accuracies were compared. The learning rates tested were 0.1, 0.01, …, 1e-05. Furthermore, 1000 iterations were used for gradient descent.

Experiment 5 Results

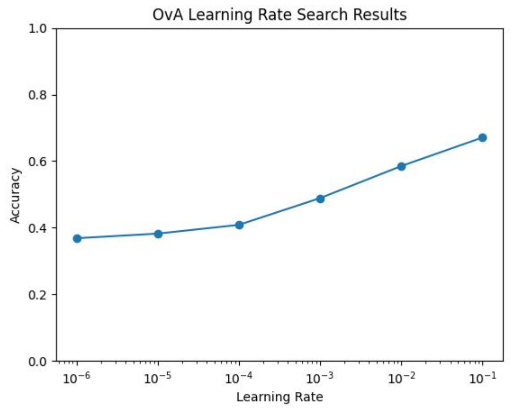

Figure 5: The learning rate search results

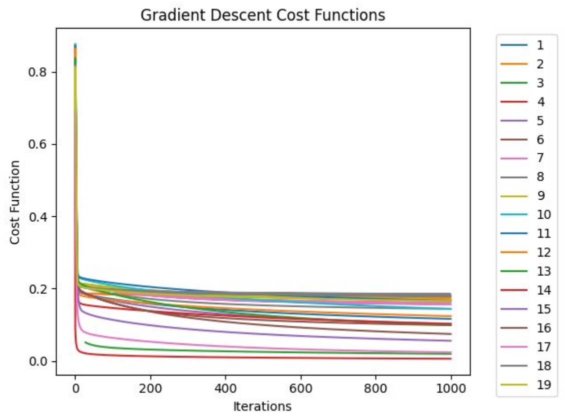

Figure 6 : The gradient descent cost functions for each of the 19 logistic regression classifiers in the OvA classifier.

Figure 5 shows the accuracies of each of the OvA classifiers trained using the learning rates listed in the previous section. All six models performed suboptimally, with the best model using a learning rate of 0.1 only getting 67.1% of predictions correct.

This poor performance may be due to the relatively low number of iterations used for gradient descent. Figure 6 shows the cost functions for each of the 19 logistic regression classifiers as it performs all 1000 iterations of gradient descent. From this graph, it is evident that each individual logistic regression classifier has a cost function that converges at different rates. In this implementation, all 19 received the same learning rate and the same number of iterations for gradient descent. It may be beneficial to fine-tune these variables for all 19 logistic regression classifiers so that their cost functions can appropriately converge to 0. This solution may solve the problem of poor accuracy.

5. Comparison with Prior Work

The implementations of Experiments 3 and 4, which are the k-nearest neighbors and decision tree experiments, were compared against the KNeighborsClassifier and DecisionTreeClassifier from the popular Python machine learning library sklearn.

The implementation of k-nearest neighbors from this project was only able to manipulate two hyperparameters: the k-value and distance metric. The KNeighborsClassifier from sklearn has many more hyperparameters to fine-tune. When testing the same best model from Experiment 3 (k=1 and metric=manhattan), sklearn's KNeighborsClassifier performed comparable to this project's implementation with an accuracy of 95.5%.

Similarly, there are several more hyperparameters to choose from in sklearn's DecisionTreeClassifier. This also performed similarly to this project's best decision tree from Experiment 4 with an accuracy of 98.0%.

6. Summary and Future Work

In conclusion, the above experiments have shown that given the right data, there are several machine learning algorithms capable of predicting human activity with great accuracy. In particular decision trees were able to correctly predict the activity being performed with 98 percent accuracy.

In the future, a more thorough evaluation of each model should be performed by using a confusion matrix. Additionally, the OvA classifier can be further fine-tuned given enough time for testing.

The ability for systems to detect a wide range of activities is certainly possible. Given the advancement of machine learning algorithms and the miniaturization of sensors, the future of human activity classification can have a greater impact across a wide variety of fields such as medicine, sports, security and elderly and infinite care.

7. Contributions

Justin Santos performed experiments 3, 4, and 5, which are the implementations and evaluations of k-nearest neighbors, decision trees, and the one-versus-all algorithm using logistic regression classifiers. Miguel Barrios performed experiments 1 and 2 implemented the neural network..

References

[1] Kerem Altun, Billur Barshan, Orkun Tunçel,"Comparative study on classifying human activities with miniature inertial and magnetic sensors,Pattern Recognition". Volume 43, Issue 10, 2010, Pages 3605-3620, ISSN 0031-3203 http://www.sciencedirect.com/science/article/pii/S0031320310001950

[2] Padraig Cunningham and Sarah Jane Delany, “k-Nearest Neighbour Classifiers”, Apr. 2007.

[3] Sahak Kaghyan and Hakob Sarukhanyan. Activity Recognition Using K-Nearest Neighbor Algorithm on Smartphone with Tri-Axial Accelerometer. *International Journal of Information Models and Analyses.* vol 1. pp.146-56. 2012.

[4] Veralia Gabriela Sánchez and Nils-Olav Skeie. “Decision Trees for Human Activity Recognition Modelling in Smart House Environments”. The University of South-Eastern Norway. Nov. 2018.

[5] Lourdes Martínez-Villaseñor, and Hiram Ponce. (2019). A concise review on sensor signal acquisition and transformation applied to human activity recognition and human–robot interaction. *International Journal of Distributed Sensor Networks, 15*(6), 155014771985398.

[6] Zunash Zaki, et. al. Logistic Regression Based Human Activities Recognition. *Journal of Mechanics of Continua and Mathematical Sciences*. vol 15. no. 4. pp. 228-46. Apr. 2020.

Appendix

Activity Number	Activity
1	sitting
2	standing
3	laying down on back
4	laying down on right side
5	ascending stairs
6	descending stairs
7	standing in elevator
8	moving in elevator
9	walking in parking lot
10	walking on a 0-degree incline treadmill at 4 km/h
11	walking on a 15-degree incline treadmill at 4 km/h
12	running on a 0-degree incline treadmill at 8 km/h
13	exercising on a stepper
14	exercising on a cross trainer
15	cycling on an exercise bike in a horizontal position
16	cycling on an exercise bike in a vertical position
17	rowing
18	jumping
19	playing basketball

Table 1: The table of activities in the dataset.