Terrain Identification using Time-Series Data

1st Arpad Voros
Electrical and Computer Engineering
North Carolina State University
Raleigh NC, USA
aavoros@ncsu.edu

2nd Josh Beerel
Electrical and Computer Engineering
North Carolina State University
Raleigh NC, USA
jbeerel@ncsu.edu

3rd Zackery Miller
Electrical and Computer Engineering
North Carolina State University
Raleigh NC, USA
zamiller@ncsu.edu

***Abstract*—This paper presents a deep learning approach to the classification of different terrain types. Time-series data of accelerometers operating in the x, y, and z axes and labeled terrain pictures are used to train a deep learning model that consists of convolutional and recurrent neural structures to predict the terrain that a lower limb prosthetic is exposed to while walking. This paper presents the preliminary implementation of this model and discusses limitations in the acquired data and initial modeling phases as well as an outline for detailing future work on this problem.**

***Index Terms*—terrain, identification, classification, CNN, RNN**

I. Methodology

Upon inspection, it was found that the timestamps for the sensor measurements was inconsistent with the timestamps of the labels. MATLAB was used to properly interpolate the data to augment the time-series component to produce a constant time-step between the two sets of data. Furthermore, the data was augmented to remove an intrinsic bias in the representation of "ground" terrain, which was found to constitute $\sim 75\%$ of the dataset; however, this was not reflected in our predictive model and will be accounted for in a future setup.

Once this stage was complete, the data was imported into a Colab Environment where the Keras deep learning framework was imported for model generation. A variety of functions native to Keras were employed for this project, including the model generator functions and the layer functions for convolutional (CNN) and recurrent neural networks (RNN). This framework enabled an intuitive implementation of a variety of different neural network models. The baseline structure of the framework will use a CNN that feeds into an RNN-structure to generate predictions [1].

Implementing a variety of different models was the strategy for determining the best structure to move forward in this project. A model evaluated was performed in regards to varying different metrics, such as model structure, time-step, filter length, and hidden layers. This model evaluation was used to select the an optimal model for the purpose of generating viable predictions.

II. Model Training and Selection

A. Model Training

For the purpose of this project, classical machine learning techniques were not considered in favor of producing a more accurate deep learning model. The general structure of the model, defined in the Methodology section, is a CNN that feeds into an RNN-structure which could be either a Long-Short Term Memory (LSTM) or a gated recurrent unit (GRU). For the training data, the interpolated data was used as the input after having been split into an $80\% - 20\%$ configuration for the training and validation sets respectively.

The first test structure was a simple 1-D CNN structure that feeds into a single LSTM layer and finally a dense layer for the output, shown in Fig. 1.

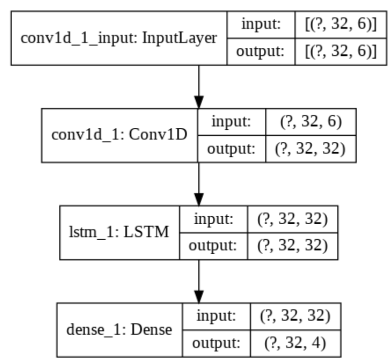

Fig. 1. Model Structure of Basic CNN-RNN Model

This initial structure was expanded upon with respect to the types of layers employed as well as the regularization methods discussed in the class - Batch Normalization and Dropout. Table 1 demonstrates the validation accuracy found for a list of models, which were used to determine the most effective model structure. Note that in the Table, a Dense layer is described as DNN, Dropout is D, and Batch Normalization is BN. Twelve different architectures were tried and evaluated.

From Table I, it can be observed that the increase in complexity of the structure results in higher accuracy at the validation stage; though, the improvement from the basic model to the more complex is not substantial with respect to the overall accuracy. Fig. 2 and 3 below show the comparison between Model 1 and the selected Model concerning their loss and validation accuracy to give a clearer picture of how these models are operating.

From the figures, the impact of the dropout layer can be seen with respect to a reduction in variation from epoch to epoch in the validation accuracy. Furthermore, the addition

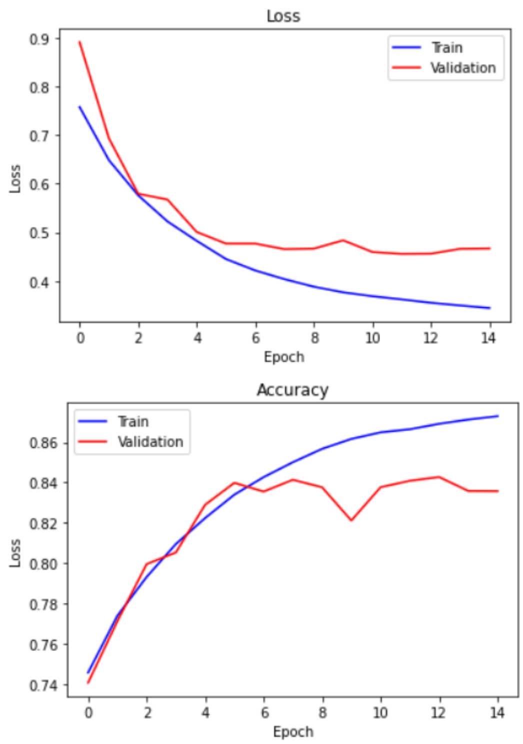

Fig. 2. Model 1 Loss and Validation Plots

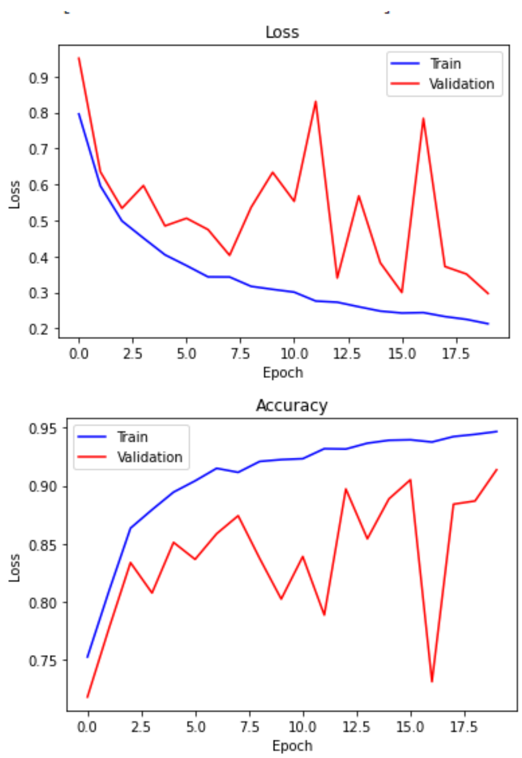

Fig. 3. Selected model Loss and Validation Plots

TABLE I
COMPARISON OF MODEL VALIDATION ACCURACY

Model No.	Model Structure	Val. Acc.
1	CNN-LSTM-DNN	0.832
2	CNN-GRU-DNN	0.839
3	CNN-LSTM-DNN(x3)	0.828
4	CNN-GRU-DNN(x3)	0.849
5	CNN-LSTM-D-DNN(x3)	0.819
6	CNN-GRU-D-DNN(x3)	0.833
7	CNN-D-LSTM-D-DNN(x3)	0.815
8	CNN-D-GRU-D-DNN(x3)	0.838
9	CNN-BN-D-LSTM-BN-D-DNN(x3)	0.836
10	CNN-BN-D-GRU-BN-D-DNN(x3)	0.844
11	CNN-BN-D(x2)-LSTM-BN-D-DNN(x3)	0.858
12	CNN-BN-D(x2)-GRU-BN-D-DNN(x3)	0.859

of a dropout layer seems to result in a faster convergence to the peak validation accuracy compared to Model 1. All of the other hyper parameters were left constant for the training of these models and are shown in Table 2.

TABLE II
STATIC HYPERPARAMETERS FOR MODEL ARCHITECTURE SEARCH

Parameter	Value
Epochs	15
Batch Size	64
Dropout Rate	0.2
Optimizer	Adam
Learning Rate	0.001
Loss Function	Cross-Entropy

B. Model Selection

Each of the 12 architectures were tested using 16 unique instances, constituting of varying number of time-steps, hidden layers, kernel sizes, etc., and were trained using the static hyperparameters outlined in Table II. The final and max categorical and validation accuracy metrics were averaged and recorded in Table I. This data was informative with regards to architecture selection for developing an optimal model for the given application. As a result, a model with three convolution layers, followed by batch normalization & dropout, GRU, and three final dense layers (CNN-BN-D(x3)-GRU-D-DNN(x3)) was developed. It is noted that the validation accuracy of this model trends upwards in a sharp oscillating fashion. This behavior is assumed to be, in part, due to the validation set format. The validation set is taken from the last 20% of an input data set that is heavily skewed towards flat surface walking movement. The section of data used for validation includes the small variance in categories of movement by the test subject at the end of the input data sample. The second iteration of this project will input balanced data and is predicted to render smoother curves for validation accuracy and loss. The evaluation of this model is discussed in the following section on the provided test data to determine how effective it is at identifying terrain based on new data. A diagram of this model is shown below.

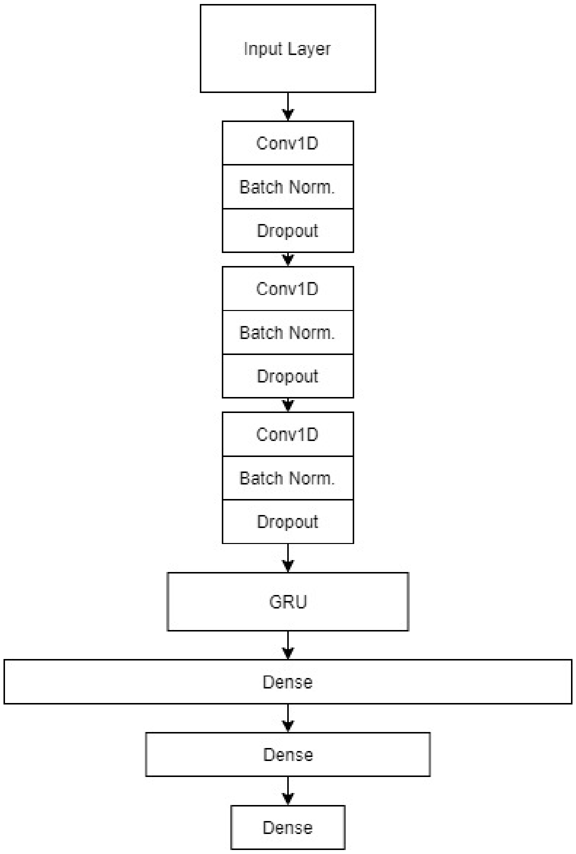

Fig. 4. Selected Model Architecture

III. Evaluation

The selected model produced a maximum accuracy of 92% and an F1 score of 0.838. Predictions were made from the resultant model which outputs a 1×4 array of probabilities for each category for each timestamp. This data was transformed into a $1\times n$ matrix that list the overall prediction per timestamp given the probability matrix, where n is the number of timestamps sampled in the prediction. Predictions were made for subjects 9-12 and the results are presented below.

Figures 5-7 show predicted transitions that each subject has made over time. You can see that with subjects 11-12, the subjects started walking on the ground, then down the stairs, and onto grass before walking back up the stairs again. It's evident that the model isn't fully certain on whether subject 10 is walking on grass or ground, as it switches between the two. Furthermore, anytime a transition is made from ground to grass, this rapid oscillation between the two categories occur. With the exception of this noise, the predictions show reasonable movement with each subject within the duration of their individual test.

References

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, p. 3–11, Mar 2019.

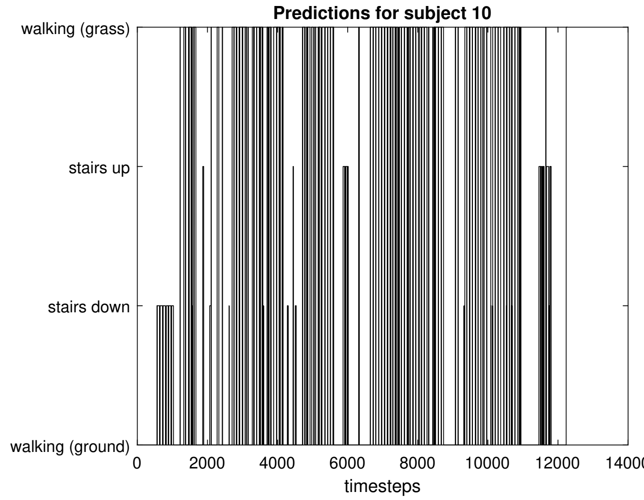

Fig. 5. Subject 10 predictions

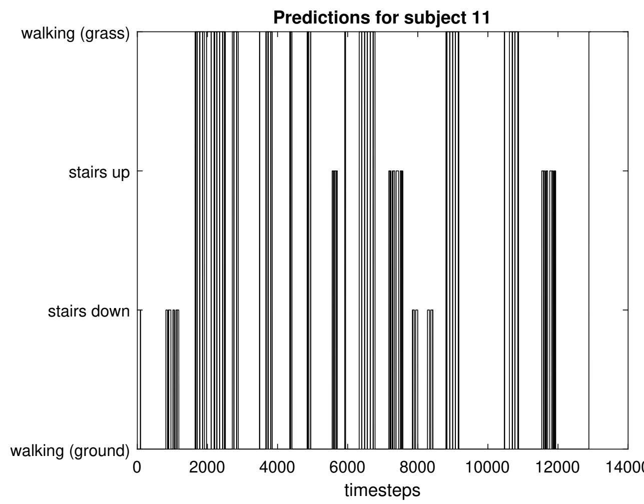

Fig. 6. Subject 11 predictions

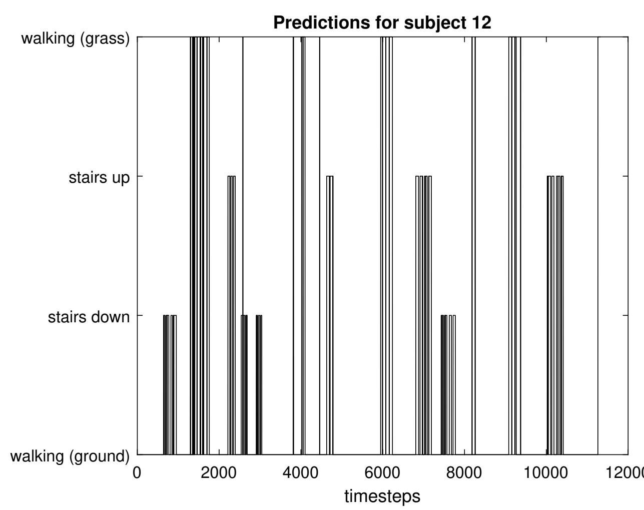

Fig. 7. Subject 12 predictions