EESEC 440: English for Electrical & Electronics Engineering*

**on Deep Learning field of study.*

{M. N. Acar, I. Avsar, Y. Akyar, D. Aladag, F. Ayas, E. Ayaz, Y. Borta, I. Calik, C. F. Canbaz, A. Cantepe, R. Celayir, F. Coskun, E. Dabag, G. Delal, M. Delibas, S. Demir, S. Derekoy, Z. Elbir, M. A. Ersoy, R. Gedik, A. O. Gur, O. Inan, M. A. Iskefyeli, H. Kabaktepe, B. Karaca, N. Koban, R. Namdar, H. H. Olcer, E. Orhan, S. Ozcelik, A. Ozcilingir, I. Ozer, ...}, {I. Acikgoz, A. Aladag, T. Albay, O. F. Alniak, M. B. Araz, M. O. Atalay, B. C. Balci, C. Bayraktar, Y. Bozkurt, M. A. E. Demir, E. G. Duz, A. Erdagli, N. Esin, ...}, M. T. Koroglu†
Electrical and Computer Engineering, Gumushane University, Gumushane, TURKEY
†*m.tahakoroglu@gumushane.edu.tr*

***Abstract*—This document is a report gradually formed by the instructor of EESEC 440 course that is taught in Spring 2021 Semester in Gumushane University. The instructor thought it would be the best to assign a particular topic to each student depending on the individual's research interest, but the vast number of students (35 + 45) made this option not possible. Therefore, a discussion is made between the students and the instructor in the first class to determine a common topic for the course. Due to diversity of student interests on different fields of study in Electrical & Computer Engineering, the instructor decided[1] to go with Deep Learning, which gained enormous importance in the last decade. EESEC 440 context is closely related to the materials presented by Dr Gokhan Cetin in EESEC 422 Artificial Intelligence course[2] given in Spring 2021 Semester.**

***Index Terms*—Deep Learning (DL), Artificial Intelligence (AI)**

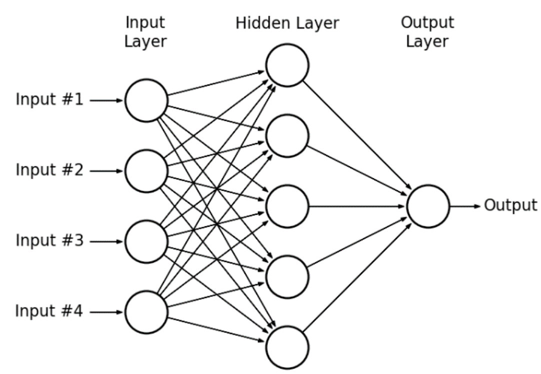

Fig. 1. A multi input single output MLP.

I. Artificial Neural Networks (ANN)

Artificial neural networks are powerful tools in building complex mathematical relationship between input-output data. The computational power rise that came with Graphical Processing Units (GPU) in the last decade enabled ANN to solve many engineering problems in various fields of study (e.g., Computer Vision, speech recognition, Natural Language Processing, power transfer [1], etc). There are different flavors of ANN designed for different applications. Multi-layer Perceptron (MLP) is the most common type of network employed in regression and classification problems, which are two major functions of a ANN.

A. Multi-layer Perceptron (MLP)

Today, we will look at two examples of MLP, one for regression and the other for classification, both supervised learning. Please see Fig. 1 for an example MLP network.

B. Convolutional Neural Networks (CNN)

In weeks 3-4-5, we will look at a classification example on images as can be seen in Fig. 2. Cats-dogs image repository on Kaggle. We will follow the tutorial given in [2].

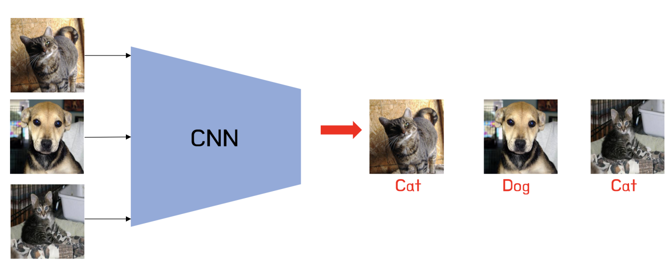

Fig. 2. Classification of cat dog images via CNN.

Mentioned absolute positioning methods are effective to localize users and provide long-term solutions. However, the infrastructure is high-cost and entails considerable labor work for setting up an LPS and maintenance. For this reason, cheaper and more practical solutions are sought. Self-contained methods compute positioning information on their

[1]Despite this mandatory decision, many of the students seemed to agree with the topic

[2]Many of EESEC 440 students are also enrolled in EESEC 422 course. Among the ones who are not enrolled in EESEC 422, a considerable amount have already taken the EESEC 422 in previous years from Dr Recep Cakmak.

own once initialized externally. Light-weight, low-cost and small-size MEMS Inertial Measurement Units (IMU) enabled capturing motion data of humans and by using dead-reckoning (DR) principle, users can be tracked for all time. Unfortunately, MEMS sensors suffer from time-variant bias and severe noise that results in completely useless trajectories in a few seconds due to accumulation of errors [3], [4]. Researchers introduced heuristic corrections such as the zero-velocity update (ZUPT) [5]–[9] and/or Heuristic Drift Reduction (HDR) [10]–[12] to compensate for positioning errors as much as possible. In particular, ZUPT helps the pedestrian inertial navigation system (INS) to estimate the traveled distance accurately while motion direction estimation issues are handled via HDR. Magnetometers can also be employed to eliminate heading errors [13], especially for outdoor applications.

II. Related Work

While a vehicle or pedestrian can move in any direction theoretically, the trajectories form certain patterns (e.g., connected straight line segments) even in wide open spaces. This fact makes topological representation of maps very powerful in positioning and navigation applications. Instead of modeling all walkable/drivable areas and regions on the map, the most descriptive information is kept in terms of nodes and edges. Topological maps are used by Kuddus et al. [14] and Greenfeld et al. [15] in GPS localization to improve the accuracy via geometric map-matching methods such as point-to-point or point-to-curve [14]–[16]. Gillieron et al. [17] and Spassov et al. [18] referred to topogical representation of maps as link/node model[3] and used it for indoors to perform the same type of map-matching on pedestrian INS trajectories. Borenstein et al. accessed open street maps (OSM) to derive angle of edges and compared the extracted information with actual HDR compensated trajectory to achieve map-matching for outdoors [16]. In the absence of initialization, the problem becomes localization of the user because the alignment of the trajectory is not known. Gupta et al. used solely edge angles of navigable paths on a Building Information Model (BIM) [23] and transport layer on Geographic Information Systems (GIS) [24] to develop a probabilistic graph traversal strategy for real-time localization. Chen et al. proposed graph neural networks (GNN) for robot agent localization in a topological map [25].

III. Problem Statement

We aim to achieve pedestrian localization on routing graphs without initial coordinates and motion direction. For trajectories that form topological connections, after sufficient number of node traversals, the shape can be fit on the map. Fig. 3 shows a pedestrian INS trajectory (red solid line) generated at a university gathering place where dashed lines are the edges that represent walkable paths and the intersections are the nodes that stand for junctions.

[3]*Routing graphs* [19]–[21] term is interchangeably used with *link/node model* [17], [18], [22] to refer to topological maps in the context of positioning.

Fig. 3. The outdoor map can be represented with a link/node model as pedestrians prefer to walk on edges instead of random trajectories. The problem becomes localization instead of map-matching in the case of unknown initial coordinates and heading.

The trajectory suffers from some positioning errors on edges and at turning points. Also edge length accuracy is not perfect despite ZUPT corrections. By using HDR algorithm [10], [11] and magnetometers [13], heading angles can be fixed to obtain straight segments. Additionally, adaptive ZUPT thresholding approach can produce highly accurate edge distances [26], [27]. Eventually, a reliable sub-graph can be extracted from error-prone trajectory. Thereby, the localization problem can be posed as a sub-graph traversal problem similar to [23]–[25]. Instead of using a generative method as in [23], [24], we propose a discriminative method based on deep learning similar to [25].

IV. Methodology

A. Features

Instead of using actual node coordinates to represent trajectories (which is not possible with uninitialized relative positioning systems), angles or edge-lengths can be used alternatively [17], [18]. Angles describe how two adjacent edges are connected and can be computed with analytic geometry of nodes or edges. Edge length is the actual length for every link on the trajectory and will be obtained directly from the map database in this section. A trajectory with n nodes can be represented with n-2 angles and n-1 lengths placed in a ordered sequence as can be seen in Fig. 4. Here we use ϕ to refer to angle and l to represent edge length.

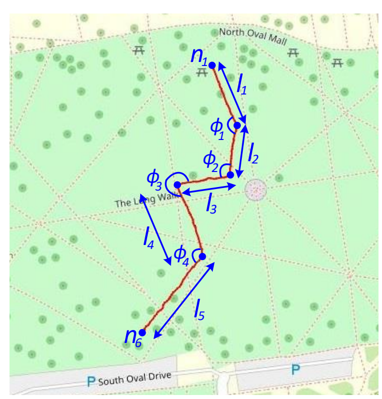

Fig. 4. Feature extraction from the sample trajectory given in Fig. 3.

B. Neural Machine Translation (NMT) Network

Based on the format of input and output training data, the sequence-to-sequence model is selected that proved its success in machine translation, text summarization and image captioning applications [28], [29]. As can be seen Fig. 5, this model takes a sequence of items (e.g., words, letters) and produces another sequence with encoder-decoder mechanism where encoder and decoder are both Recurrent Neural Networks (RNN). Aside from adopting the vanilla sequence-to-sequence model, we employed an attention mechanism over the encoder Long Short Term Memory (LSTM) states [30], [31]. This treatment provides a random access memory to source hidden states that can be accessed throughout the translation process [32].

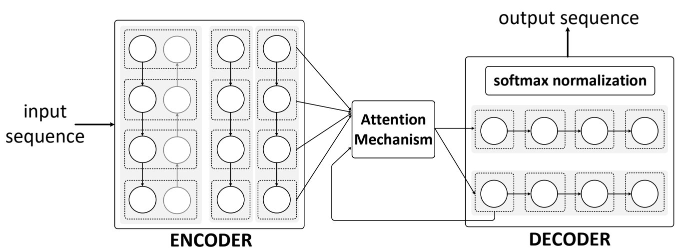

Fig. 5. Neural Machine Translation Network with attention mechanism.

In this paper, we adopted an NMT English2German translation model [33] and changed it as necessary. The training mini-batch is updated to use all the input training data within a single training epoch. Also the testing mechanism is modified such that we can feed testing input to the trained NMT model and save both testing and predicted output to compute prediction accuracy rate in future steps.

C. Input Training Data and Preprocessing

Input training data are ordered sequences of features extracted from trajectories consisting of n nodes. Three types of input training data sequence can be formed. The first one is angle sequence, the second one is length sequence and the last one is angle-length sequence, which combines both angle and edge length, as can be seen in TABLE I.

TABLE I
INPUT TRAINING DATA SEQUENCE FORMATS

ϕ_1	ϕ_2	$\ldots$	ϕ_{n-3}	ϕ_{n-2}

l_1	l_2	$\ldots$	l_{n-2}	l_{n-1}

l_1	ϕ_1	l_2	ϕ_2	$\ldots$	l_{n-2}	ϕ_{n-2}	l_{n-1}	$NULL$

The range of angles between two adjacent edges is $[0, \pi]$. The first step is reducing the computational complexity by quantizing the angle into 72 bins, that corresponds to 2.5^o bin size [24].

$$\phi_{quantized} = \lfloor \phi_{original}/2.5 \rceil \tag{1}$$

The second step is to map the quantized angle values between $[0, 1]$, i.e., normalization, and keep three decimal digits. The latter extends 72 bins to 1001 bins, which is very effective in improving prediction accuracy:

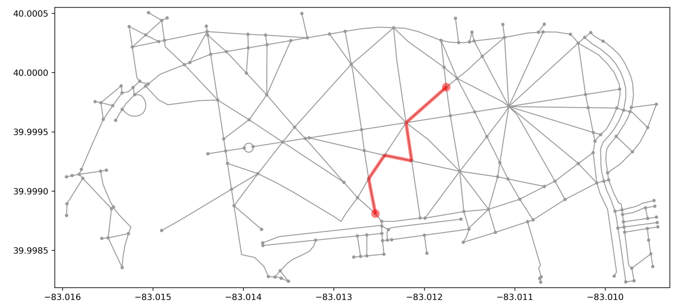

Fig. 6. Open street map (OSM) of the gathering place and the corresponding localized trajectory (see Fig. 3 for the actual uninitialized trajectory).

$$\phi_{normalized} = \frac{\phi_{original} - \phi_{min}}{\phi_{max} - \phi_{min}} \tag{2}$$

We perform preprocessing on unique edge lengths as well. Similar to the angles, the first step is to build a dictionary by relabeling these length values with ordered integers such as 1, 2, 3 and so on. The second step is normalization while keeping three decimal digits to have the same resolution:

$$l_{normalized} = \frac{l_{original} - l_{min}}{l_{max} - l_{min}} \tag{3}$$

D. Target Data and Output

The target data (i.e., ground truth labels) is a sequence of node coordinates appearing in the order of pedestrian traversal. Each node on OSM is associated with a unique ID that is a scalar. For convenience, a single-row sequence consisting of node IDs are relabeled with integers such as 1, 2, 3 and so on.

V. Experiments and Results

In order to learn the motion patterns on the topological map, the trajectories are generated randomly and nonredundantly on the walkable paths of the gathering place shown in Fig. 3. The corresponding open street map (OSM) and localized trajectory is shown in Fig. 6.

The accuracy of localization is inherently proportional to the number of traversed nodes; therefore, accuracy vs. traversed nodes results are illustrated in Fig. 7 using only (i) angles, and only (ii) edge lengths. The performance of localization with fixed six node traversals is elaborately shown in Fig. 7 (iii). *All nodes* refer to node-wise cumulative accuracy rate while *only end node* denotes the accuracy based on the correctness of the terminating node (i.e., if the pedestrian finished at the correct junction). The accuracy passes 95% when number of traversed nodes become only 6 in Fig. 7 (i) and (ii). The effect of normalization can be seen in Fig. 7 (iii).

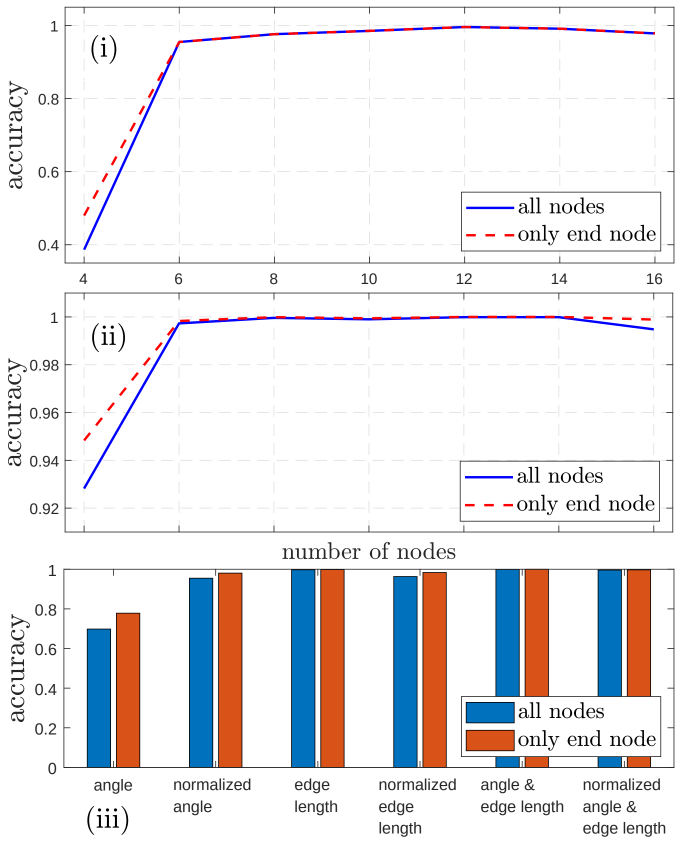

Fig. 7. Localization accuracy vs. traversed nodes for using only (i) angles, (ii) edge lengths. (iii) Localization accuracy vs. different feature and preprocessing settings when the number of traversed nodes is only six.

Conclusions and Future Work

In this paper, we adopted a neural network structure that is originally designed for language translation. After proper modifications on the structure and preprocessing on input-output data, proposed method can localize a user on a topological map without initial position and heading information by traversing only six nodes with an accuracy of 95%. Current goal is to collect extensive relative-positioning data with cars and pedestrians to further validate the developed method. Integration of the presented localization solution into a general relative positioning method where the user is not restricted to prescribed paths is an ongoing study.

References

[1] T. Guillod, P. Papamanolis, and J. W. Kolar, "Artificial neural network (ann) based fast and accurate inductor modeling and design," *IEEE Open Journal of Power Electronics*, vol. 1, pp. 284–299, 2020.

[2] "How to classify photos of dogs and cats (with 97% accuracy)." https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/. Accessed: 2021-3-10.

[3] O. J. Woodman, "An introduction to inertial navigation," Tech. Rep. UCAM-CL-TR-696, University of Cambridge, Computer Lab., Aug. 2007.

[4] P. Aggarwal, Z. Syed, A. Noreldin, and N. El-Sheimy, *MEMS-based Integrated Navigation*. GNSS technology and applications series, Artech House, 2010.

[5] S. Godha and G. Lachapelle, "Foot mounted inertial system for pedestrian navigation," *Measuserement Science & Technology*, vol. 19, p. 075202, May 2008.

[6] A. R. Jimenez Ruiz, F. Seco Granja, J. C. Prieto Honorato, and J. I. Guevara Rosas, "Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, pp. 178–189, Jan 2012.

[7] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, pp. 38–46, Nov. 2005.

[8] C. Fischer, P. Talkad Sukumar, and M. Hazas, "Tutorial: Implementing a pedestrian tracker using inertial sensors," *IEEE Pervasive Computing*, vol. 12, pp. 17–27, April 2013.

[9] J. Nilsson, A. K. Gupta, and P. Händel, "Foot-mounted inertial navigation made easy," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 24–29, Oct 2014.

[10] J. Borenstein, L. Ojeda, and S. Kwanmuang, "Heuristic reduction of gyro drift for personnel tracking systems," *Journal of Navigation*, vol. 62, no. 1, p. 41–58, 2009.

[11] A. R. Jimenez, F. Seco, J. C. Prieto, and J. Guevara, "Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU," in 7^{th} *WPNC*, pp. 135–143, March 2010.

[12] Y. Li, Q. Song, M. Ma, and Y. Gu, "Near real time heading drift correction for indoor pedestrian tracking based on sequence detection," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–4, Oct 2016.

[13] A. R. Jimenez, F. Seco, F. Zampella, J. Prieto, and J. Guevara, "Improved heuristic drift elimination with magnetically-aided dominant directions (MiHDE) for pedestrian navigation in complex buildings," *Journal of Location Based Services*, vol. 6, no. 3, pp. 186–210, 2012.

[14] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 5, pp. 312 – 328, 2007.

[15] J. S. Greenfeld, "Matching gps observations to locations on a digital map," *Proceedings of the* 81^{th} *Annual Meeting of the Transportation Research Board*, pp. 1–13, Jan 2002.

[16] P. Aggarwal, D. Thomas, L. Ojeda, and J. Borenstein, "Map matching and heuristic elimination of gyro drift for personal navigation systems in GPS-denied conditions," *Measurement Science and Technology*, vol. 22, p. 025205, Jan 2011.

[17] P.-Y. Gilliéron, I. Spassov, and B. Merminod, "Indoor navigation enhanced by map-matching," *European Journal of Navigation*, vol. 3, no. 3, 2005.

[18] I. Spassov, M. Bierlaire, and B. Merminod, "Map matching for pedestrians via bayesian inference," in *European Navigation Conference - Global Navigation Satellite Systems*, May 2006.

[19] T. Willemsen, F. Keller, and H. Sternberg, "A topological approach with MEMS in smartphones based on routing-graph," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–6, Oct 2015.

[20] F. T. Alaoui, D. Betaille, and V. Renaudin, "A multi-hypothesis particle filtering approach for pedestrian dead reckoning," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, Oct 2016.

[21] F. Taia Alaoui, D. Betaille, and V. Renaudin, "Pedestrian dead reckoning navigation with the help of A^*-based routing graphs in large unconstrained spaces," *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–10, 07 2017.

[22] J. Jiang, H. Le, and S. Shie, "Lightweight topological-based map matching for indoor navigation," in *2016* 30^{th} *International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 908–913, March 2016.

[23] A. Yilmaz and A. Gupta, "Indoor positioning using visual and inertial sensors," in *2016 IEEE SENSORS*, pp. 1–3, Oct 2016.

[24] A. Gupta and A. Yilmaz, "Ubiquitous real-time geo-spatial localization," in *Proceedings of the Eighth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, ISA '16, (New York, NY, USA), pp. 1–10, ACM, 2016.

[25] K. Chen, J. P. de Vicente, G. Sepulveda, F. Xia, A. Soto, M. Vazquez, and S. Savarese, "A behavioral approach to visual navigation with graph localization networks," *arXiv preprint arXiv:1903.00445*, 2019.

[26] B. Wagstaff and J. Kelly, "LSTM-based zero-velocity detection for robust inertial navigation," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, Sep. 2018.
[27] R. Zhang, H. Yang, F. Höflinger, and L. M. Reindl, "Adaptive Zero Velocity Update Based on Velocity Classification for Pedestrian Tracking," *IEEE Sensors Journal*, vol. 17, pp. 2137–2145, April 2017.
[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3104–3112, Curran Associates, Inc., 2014.
[29] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, Oct. 2014.
[30] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," in *Advances in neural information processing systems*, pp. 2773–2781, 2015.
[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.
[32] R. Pryzant, "Evaluating tensorflow," 2017.
[33] N. Tomic, "Neural machine translator for english2german translation."