Modeling Robotic Surgery Predictions: Write-Up

Meftah, Morteza M.D; Waren, Daniel M.D; Bosco, Joseph A. IV;
Di Gangi, Catherine; Watson, Cody Ph.D

March 19, 2024

DISCLAIMER: This is a VERY rough draft of the write-up. It is meant to demonstrate the work I have done on the project in a way that you don't need to read any code. Document written by Jack Bosco. Code for the data analysis, modeling, and visualization are also written by Jack Bosco.

1 Methods

Overview

Analysis of pre-operative planning and report screenings from 626 TKR cases was used to establish the NYU method for approximating optimal post-resection tibial and femoral coronal alignments based on the pre-operative tibial and femoral coronal alignments. To establish a unified input format, femoral and tibial coronal alignments are translated into MPTA and LDFA measurements, respectively. MPTA and LDFA were chosen since they allow for easy computation of aHKA and JLO values to visualize the knee morphology through CPAK classification [5]. After calculating the measurements, MPTA and LDFA values are fed into a min-max scaler transformation to normalize the input and output values within the range [-1, 1] [6]. The last pretraining step splits the normalized data into testing, training and validation sets of sizes 30%, 66.5% and 3.5% respectively with random seed 42. In training the model runs stochastic gradient descent, looping until a maximum number of repetitions is reached or 10 repetitions are completed with no change in error. After it is finished training, the model is evaluated using the mean squared error between the de-normalized predicted aHKA and target aHKA values from the testing dataset.

Pre-Processing

The pre-processing step involves all the actions performed on the raw dataset before the model can be trained. In this experiment, tibial and femoral coronal alignments of the knee region are translated into MPTA and LDFA measurements, respectively. The MPTA measurement is a representation of the tibial coronal alignment such that perfectly aligned tibial coronal angles map to 90° and the equations aHKA = MPTA − LDFA and JLO = MPTA + LDFA satisfy the CPAK morphologies defined by [5]. Similarly, the LDFA measurement is a representation of the femoral coronal alignment such that perfectly aligned femoral coronal angles map to 90° and the equations aHKA = MPTA − LDFA and JLO = MPTA + LDFA give the CPAK morphologies defined by MacDessi et. al.[5]. For the input space X, pre-operative MPTA and LDFA angles are combined into vectors $x_i = [\text{preop MPTA}_i, \text{preop LDFA}_i]$ where i denotes the REDCAP ID number. The same is done for the output space Y : $y_i = [\text{planned MPTA}_i, \text{planned LDFA}_i]$. To train and evaluate the model, we split the input and output data into testing, training and validation sets of sizes 30%, 66.5% and 3.5% respectively. Each split contains an equal number of pre-op and planned values, linked by case through the REDCAP ID.

Next, two normalizers are defined: norm_Y and norm_Y. They use the min max scaler algorithm defined in the scikit-learn library [7] to normalize the values of the input and output by scaling them within the range [-1, 1]. norm_x is fitted to the training subset of X, $norm_Y$ is fitted to the training subset of Y. After fitting, the normalizer coefficients are saved so that individual [MPTA, LDFA] samples can be normalized and de-normalized deterministically.

Training

For the NYU method a multilayer perceptron (MLP) model, also known as a neural network, is trained to determine the regression line of best fit [3]. An MLP architecture for the model was chosen based on the following factors: the size of the dataset, the input dimensions and the output dimensions. Since the size of the dataset is small (626 cases), a more robust deep learning architecture is not an option. The MLP architecture we used had three hidden layers of sizes 16, 8 and 4 (Figure 1), making it large enough to learn feature representations of the input and small enough to converge on a small sample size [2]. We used stochastic gradient descent as the optimizer, set the random seed to 42, and used hyperbolic tangent as our activation function since its range (-1,1) fits well within the range of the normalizer [-1,1] [11, 3]. Other hyperparameters are set to their default value as per the scikit-learn documentation [1, 9, 7, 4]. The input dimension (2) and output dimension (2) disqualified

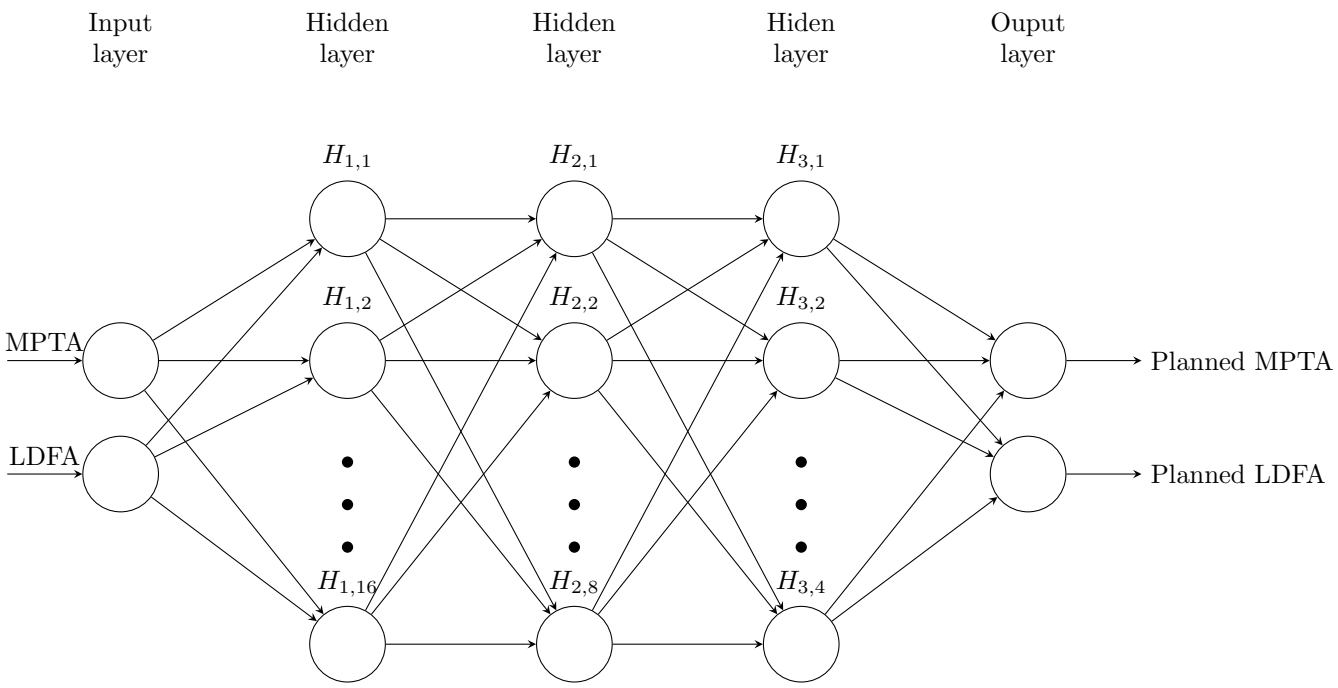


Figure 1: Diagram of the MLP model architecture

the option of using traditional support vector machines for this model. Multi-output SVM regressors are an option, however these work by combining two separate regressors [10], one to predict MPTA and one to predict LDFA, and would therefore blind to the relationship between pre-operative and post-operative aHKA values.

Evaluation

To measure the accuracy of the model we calculated and compared the aHKA values for pre-operative and planned knee alignments. To do so, we first took the following difference between non-normalized MPTA and LDFA patterns within the testing sets: X_{test}, Y_{test}:

$$X_{\text{test, aHKA}} = \{\text{MPTA} - \text{LDFA} | [\text{MPTA}, \text{LDFA}] \in X_{\text{test}}\}$$

$$Y_{\text{test, aHKA}} = \{\text{MPTA} - \text{LDFA} | [\text{MPTA}, \text{LDFA}] \in Y_{\text{test}}\}$$

Then the model predicts planned LDFA and MPTA values in the testing set (Y_{pred}). The model predicts the normalized the input ($\text{norm}_x(x_i)$), running the normalized input through the model ($\text{predict}(\text{norm}_x(x_i))$), and de-normalizing the predicted planned alignmnets (norm_y^{-1}):

$$Y_{\text{pred}} = \{\text{norm}_y^{-1}(\text{predict}(\text{norm}_x(x_i))) | x_i \in X_{\text{test}}\}$$

To compare our predictions with $Y_{\text{test, aHKA}}$, we evaluate $Y_{\text{pred, aHKA}}$ in the same manner as $Y_{\text{test, aHKA}}$:

$$Y_{\text{pred, aHKA}} = \{\text{MPTA} - \text{LDFA} | [\text{MPTA}, \text{LDFA}] \in Y_{\text{test}}\}$$

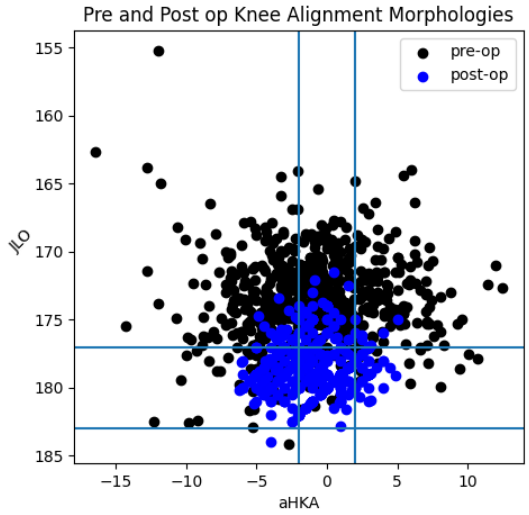
Since the testing set is split from the training set the model has never seen the patterns in X_{test} or Y_{test} before. By doing so, we get a preliminary understanding of how this model may perform in the field. As the final step, we take the mean squared error between the targets $Y_{\text{test, aHKA}}$ and the predictions $Y_{\text{pred, aHKA}}$ to evaluate the model's accuracy.

2 Results

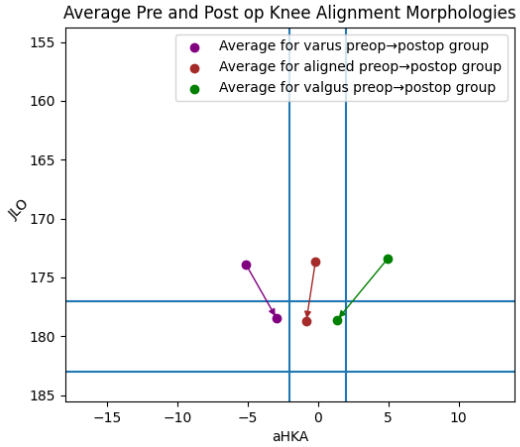
Training The model converged early, halting after training for 86with a batch size of 4patterns (Figure 4). After about 6epochs, the mean squared error between normalized input/target patterns begins to converge. The model takes less than 1 minute to converge with no GPU, however it is important to note this is with a small training set of 417 patterns. Nonetheless, with the simple feed-forward neural net architecture chosen for this model, it is expected that the model will train relatively quickly.

Shape

The regression this model produces is in-line with the hypothesized change in pre-op to planned aHKA. In Figure 2a, the pre-operative and post-operative aHKA and JLO values from the initial dataset are shown.



(a) Pre-op vs. Planned value distributions showing CPAK morpholigies.



(b) Average change from pre-op to planning stage by aHKA morphology

Figure 2: Visualization of dataset

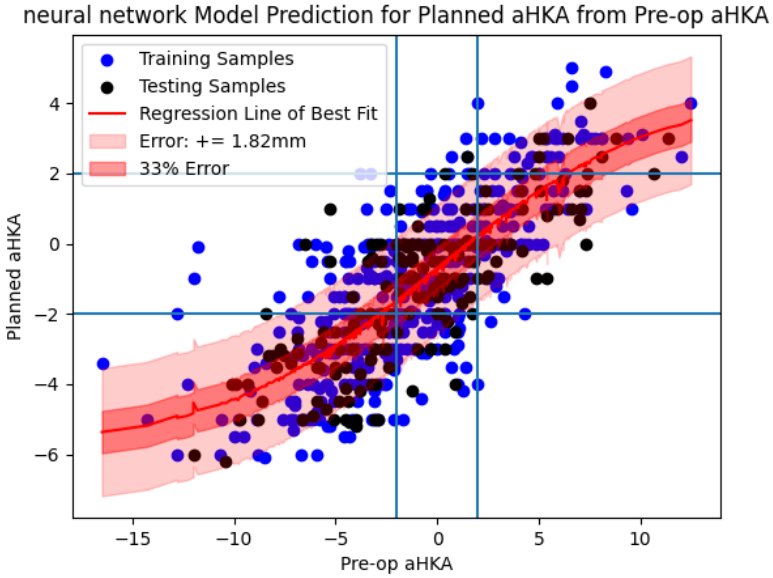


Figure 3: Visualization of the regression line of best fit produced by the MLP model

The pre-operative values were more varied than their planned counterparts and the vertical and horizontal bars separate the different knee alignment morphologies [5]. Splitting the pre-operative values by aHKA into varus, valgus and aligned groups, we can view the average change for each group from pre-op to planning stage (Figure 2b). For each group, the JLO tends to move in the same direction. More interestingly, however, is the way aHKA changes by group: on average, varus morphologies tend to stay varus and aligned morphologies tend to stay aligned but valgus morphologies are planned to be corrected to aligned morphologies. Looking at the regression produced from the model (Figure 3), we see the same pattern. Blue dots represent the training and validation set while block dots represent the testing set. The middle-left square is hardly intersected by the regression, indicating the model rarely predicts aligned morphologies from varus morphologies. The middle-right square is intersected by the regression, however, indicating the model sometimes predicts aligned morephologies from valgus morphologies.

Accuracy

Evaluative statistics demonstrate the NYU method for planning knee morphologies is more accurate than making predictions about the mean (RMSE 1.819; Nash Sutcliffe 0.587) [8]. As the regression (Figure 3) moves towards more densely populated regions of the scatter plot, the line of best straightens out. This demonstrates

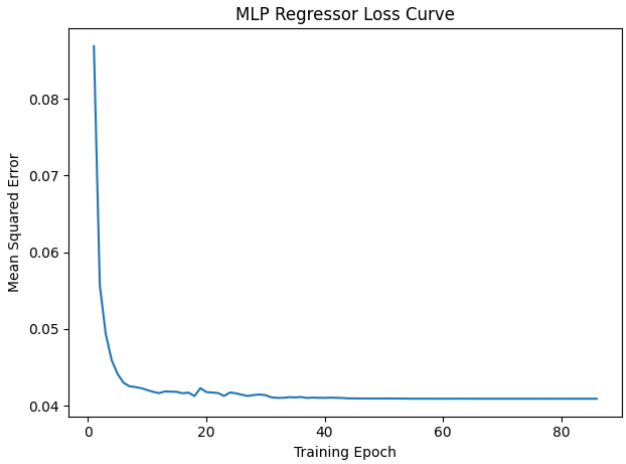


Figure 4: Loss curve for the MLP training. Mean squared error is shown for each epoch (4 repitions) from the normalized prediction/target patterns.

how the model successfully avoids overfitting so long as there are enough examples in the dataset.

Moving Forward

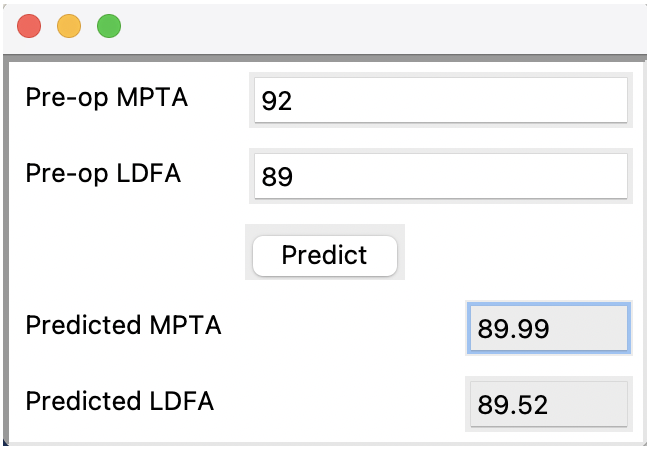


Figure 5: Graphical User Interface (GUI) with example prediction

More work needs to be done to evaluate the NYU methods usefulness for planning robotic TKR in the field. Included in this study is a codebase from which the model can be trained if suppliad appropriate data. With a pre-trained model, the NYU method is accessible through a graphical user interface (Figure 5). The GUI is brought up by running the gui.py script with a pre-trained model in the appropriate folder. With it, the NYU method can be utilized without any code.

References

[1] *1.5. Stochastic Gradient Descent — scikit-learn 1.4.1 documentation*. Mar. 2024. URL: `https://scikit-learn.org/stable/modules/sgd.html`.

[2] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256. URL: `https://proceedings.mlr.press/v9/glorot10a.html`.

[3] Geoffrey E. Hinton. "Connectionist learning procedures". In: *Artificial Intelligence* 40.1 (1989), pp. 185–234. ISSN: 0004-3702. DOI: `https://doi.org/10.1016/0004-3702(89)90049-0`. URL: `https://www.sciencedirect.com/science/article/pii/0004370289900490`.

[4] Samy Jelassi and Yuanzhi Li. "Towards understanding how momentum improves generalization in deep learning". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, July 2022, pp. 9965–10040. URL: https://proceedings.mlr.press/v162/jelassi22a.html.

[5] Samuel J MacDessi et al. "Coronal Plane Alignment of the Knee (CPAK) classification". en. In: *Bone Joint J.* 103-B.2 (Feb. 2021), pp. 329–337.

[6] S. Gopal Krishna Patro and Kishore Kumar Sahu. *Normalization: A Preprocessing Stage*. 2015. arXiv: 1503.06462 [cs.OH].

[7] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[8] Axel Ritter and Rafael Muñoz-Carpena. "Performance evaluation of hydrological models: Statistical significance for reducing subjectivity in goodness-of-fit assessments". In: *Journal of Hydrology* 480 (2013), pp. 33–45. ISSN: 0022-1694. DOI: https://doi.org/10.1016/j.jhydrol.2012.12.004. URL: https://www.sciencedirect.com/science/article/pii/S0022169412010608.

[9] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM". In: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. Corvalis, Oregon, USA: Association for Computing Machinery, 2007, pp. 807–814. ISBN: 9781595937933. DOI: 10.1145/1273496.1273598. URL: https://doi.org/10.1145/1273496.1273598.

[10] Nguyen Khoa Tran, Laura C. Kühle, and Gunnar W. Klau. "A critical review of multi-output support vector regression". In: *Pattern Recognition Letters* 178 (2024), pp. 69–75. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2023.12.007. URL: https://www.sciencedirect.com/science/article/pii/S0167865523003549.

[11] Tong Zhang. "Solving large scale linear prediction problems using stochastic gradient descent algorithms". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: Association for Computing Machinery, 2004, p. 116. ISBN: 1581138385. DOI: 10.1145/1015330.1015332. URL: https://doi.org/10.1145/1015330.1015332.