Towards a Better Scoring

Ivan J. Tashev*, R. Michael Winters*, Yu-Te Wang*, David Johnston*, Justin Estepp†, Nathaniel Bridges†
*Microsoft Research Lab – Redmond, WA
†Air Force Research Laboratory – AFRL, Dayton, OH

***Abstract*—In this paper, we propose a novel method to calculate trainee performance scores in a flight simulator environment using flight simulator logs. Our approach improves upon the existing scoring system designed in the AFRL by better fitting scores into the existing model of the training process.**

I. Introduction

In this paper, we propose a new method to calculate trainee performance scores in a flight simulator environment using flight simulator logs. The training process consists of multiple sessions of straight-line flight with two flavors: 1) a "Straight and Level" flight where the trainee should maintain constant speed, altitude, and course; and 2) a "Glideslope" flight where the trainee should maintain straight-line flight towards the beginning of the runway (i.e., course) with constant speed. The duration of each session is around 2-3 minutes. The performance of trainees in each session should be evaluated and scored with a positive real number between 0 and 1 for evaluation and optimization of the training process.

Our proposed approach improves upon the existing scoring system designed by AFRL by better fitting scores into the existing model of the training process. In the existing scoring system, information is obtained from flight simulator logs including normalized deviations from prescribed airspeed σ_{Vnorm}, prescribed altitude σ_{Anorm}, prescribed course σ_{Lnorm}, and direction to the beginning of the runway σ_{Gnorm}. The performance score is computed as follows:

$$S = \left| \begin{array}{l} \frac{((1-\sigma_{Vnorm})+(1-\sigma_{Anorm})+(1-\sigma_{Lnorm}))}{3} \\ \frac{((1-\sigma_{Vnorm})+(1-\sigma_{Anorm})+(1-\sigma_{Gnorm}))}{3} \end{array} \right. \quad (1)$$

In this equation, the upper line represents straight-and-level flight, while the second line represents glide-slope flight.

The main problem of the current scoring system is that it assigns negative scores when trainees are not experienced enough to maintain the prescribed parameters. Internally, the training system model [1] expects positive numbers and zeroes all negative scores. Thus the usability of the current model is limited for novice trainees.

The scoring process can be generalized as a weighted sum of these four features:

$$S = \sum_{i=1}^{N} w_i \sigma_i + w_0 \quad (2)$$

The four features above are task-dependent and suitable for scoring straight-line flight only. However, three additional features can be obtained from flight simulator logs: normalized deviations of the throttle σ_{Tnorm}, and the normalized deviations of the stick in two axes σ_{Xnorm} and σ_{Ynorm}. These features are less task-dependent and can be easily added to the generalized scoring in equation (2).

TABLE I
Correlation of the features with the ideal score.

Parameter	Corr. coeff.
σ_{Vnorm}	-0.6921
σ_{Anorm}	-0.1827
σ_{Lnorm}	-0.7237
σ_{Gnorm}	-0.5855
σ_{Tnorm}	-0.1969
σ_{Xnorm}	-0.2509
σ_{Ynorm}	-0.2235

II. Proposed scoring algorithm

We propose using ideal scores from the training process model described in [1]. Each trainee is modeled with three parameters: initial absolute skill, learning rate, and forgetting factor. Every scenario is modeled with two parameters: scenario difficulty and maximum achievable score. Variability of human performance is modeled as Gaussian noise added to the score. Based on scores from multiple subjects running sessions with scenarios with variable difficulty, the parameters above can be estimated, and ideal scores computed. We will use these scores to find a better way to combine the four features above. The correlation coefficients of these features with the model scores are shown in Table I. While the first three original features have relatively high correlation with the modeled score, for the second group of three features, the correlation is relatively low.

We propose treating the scoring as a regression machine learning problem: from a set of features (the deviations from the flight logs), compute the output value (the ideal scores). The evaluation criterion can be the root mean squared error (RMSE) of the interpolation. As a baseline, we will use estimation using Equation 1. Several machine learning techniques are under consideration:

- Linear regression, which is the straightforward estimation of the weights in equation (2) using least squares method [2].
- Support Vector Machine (SVM) in regression mode [3].
- Deep Neural Network (DNN) with a given number of layers and nodes in each layer [4].

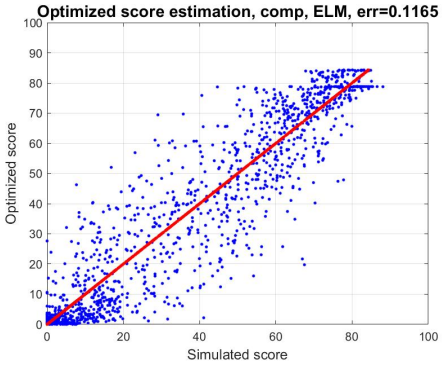

Fig. 1. Scores estimation using the original features as function of the simulated scores and ELM estimator.

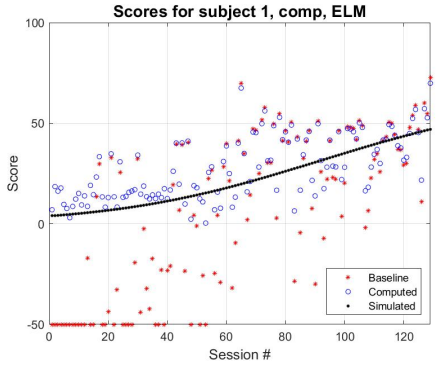

Fig. 2. Baseline, estimated, and simulated scores for one subject, ELM estimator.

TABLE II
RMSE OF THE PROPOSED APPROACHES.

Algorithm	Validation	Test
Baseline	0.5128	0.5128
Linear	0.1668	0.1952
SVM	0.1942	0.2052
DNN	0.1890	0.1950
ELM	0.1030	0.1145

TABLE III
RMSE OF DNN AND ELM WITH VARIOUS FEATURE SETS.

Feature set	Valid. DNN	Test DNN	Valid. ELM	Test ELM
Original	0.1890	0.1950	0.1030	0.1145
Orig.+contr.	0.1619	0.2002	0.3200	0.5301
Controls	0.2328	0.2694	0.5322	0.3566

- Extreme Learning Machine (ELM) in regression mode, which is a shallow and wide neural network with one hidden layer and analytic solution for the training [5].

III. Training Processes, Approaches and Strategies

To evaluate the proposed approach, we used data from 34 subjects who performed training on 11 scenarios with varying difficulty levels. The dataset consists of 1290 sessions. From the training process model [1], we retrieved 1290 ideal scores that were used as labels.

Of these 34 subjects, seven had more than 90 scores. The validation and testing datasets were selected from among them, giving us 42 combinations for training, validation, and testing datasets. The final RMSE for validation and testing was computed as a weighted average of the RMSE in each combination.

We consider three sets of features:

- The four initial features from Equation (1);
- The combination of these four plus the three task independent features;
- Only the three task independent features.

The training of the regression DNN was limited to 150 epochs with forced stopping if the results on the validation dataset did not improve for five epochs. The stochastic gradient descent algorithm was used for training with an initial learning rate of 0.001. The hyper-parameters of each approach (number of layers, number of nodes, etc.) were optimized for each regression strategy using the averaged RMSE on the validation datasets. This resulted in 16 nodes in the ELM hidden layer and three layers of 32 nodes for the regression DNN.

IV. Results

The results from all approaches using the first feature set are shown in Table II. Overall, all the regression algorithms show better accuracy than the baseline. The best performing algorithm with this feature set is the ELM, followed by the DNN. Surprisingly, the linear regression model outperforms the SVM. In Fig. 1, estimated scores are shown as a function of idealized scores. Estimated scores are positioned around simulated scores; there are no scores below zero. Fig. 2 shows baseline scores, estimated scores and scores from the model. It is clear that the estimated scores are closer to ideal scores from the model and the problem of the large number of scores below zero is resolved. Table III shows the accuracy on validation and test data sets for the ELM and DNN for all three feature sets. From these results it is clear that adding the three features did not improve accuracy on the test set, so these features do not carry enough information to be considered independently. Also noticeable is that the DNN estimator, while less accurate on first feature set, is more stable on different feature sets. Further, the ELM cannot provide good solutions for the second and third features sets, when the controls are added.

References

[1] I. Tashev, R. M. Winters, Y.-T. Wang, D. Johnston, A. Reyes, and J. Estepp, "Modeling the training process," in *2022 IEEE Research and Applications of Photonics in Defense Conference (RAPID)*, September 2022.

[2] A. S. Goldberger, "Classical linear regression," *Econometric Theory*, p. 158, 1964.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, p. 273–297, 1995.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436–444, 2015.

[5] G.-B. Huang, Q.-Y. Zhu, and S. C.-K., "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, p. 489–501, 2006.