A General Neural Architecture for Carbohydrate and Bolus Recommendations in Type 1 Diabetes Management

Jeremy Beauchamp and **Razvan Bunescu** and **Cindy Marling**[1]

Abstract. People with type 1 diabetes must constantly monitor their blood glucose levels and take actions to keep them from getting either too high or too low. Having a snack will raise blood glucose levels; however, the amount of carbohydrates that should be consumed to reach a target level depends on the recent history of blood glucose levels, meals, boluses, and the basal rate of insulin. Conversely, to lower the blood glucose level, one can administer a bolus of insulin; however, determining the right amount of insulin in the bolus can be cognitively demanding, as it depends on similar contextual factors. In this paper, we show that a generic neural architecture previously used for blood glucose prediction in a *what-if* scenario can be converted to make either carbohydrate or bolus recommendations. Initial experimental evaluations on the task of predicting carbohydrate amounts necessary to reach a target blood glucose level demonstrate the feasibility and potential of this general approach.

1 Introduction and Motivation

Type 1 diabetes is a disease in which the pancreas fails to produce insulin, which is required for blood sugar to be absorbed into cells. Without it, that blood sugar remains in the bloodstream, leading to high blood glucose levels (BGLs). In order to manage type 1 diabetes, insulin must be administered via an external source, such as injections or an insulin pump. People with type 1 diabetes also need to monitor their BGLs closely throughout the day by testing the blood acquired through fingersticks and/or by using a continuous glucose monitoring (CGM) system. If the BGL gets too high (hyperglycemia) or too low (hypoglycemia), the individual responds by eating, taking insulin, or taking some other action to help get their BGL back to within a healthy range. An issue with this, however, is that the person with diabetes must *react* to their BGL, whereas, ideally, they would be able to *proactively* control their BGL. There has been much work in the area of BGL prediction in the past ([1] and [8] for example) with the aim of enabling preemptive actions to manage BGLs before individuals experience the negative symptoms of hypoglycemia or hyperglycemia. However, individuals still need to figure out how much to eat, how much insulin to take, and what other actions they can take to prevent hypoglycemia or hyperglycemia.

The broad goal of the research presented in this paper is to essentially reverse the blood glucose prediction problem, and instead predict how many carbohydrates an individual should eat or how much insulin to administer with a bolus in order to get their BGL to the desired target. We have previously introduced in [6] an LSTM-based neural architecture that was trained such that it could answer *what-if* questions of the type "What will my BGL be in 60 minutes if I eat a snack with 30 carbs 10 minutes from now". We show that by using the BGL target as a feature and the carbohydrates or insulin as labels, a similar architecture can be trained instead to predict the number of carbohydrates that need to be consumed or the amount of insulin that needs to be delivered during the prediction window in order to reach that BGL target.

The work by Mougiakakou and Nikita [7] represents one of the first attempts to use neural networks for recommending insulin regimens and dosages. Bolus calculators were introduced as early as 2003 [11], wherein a standard formula is used to calculate the amount of bolus insulin based on parameters such as carbohydrate intake, carbohydrate-to-insulin ratio, insulin on board, and target BGL. Walsh et al. [10] discuss major sources of errors and potential targets for improvement, such as utilizing the massive quantities of clinical data being collected by bolus advisors. As observed by Cappon et al. in [2], the standard formula approach ignores potentially useful preprandial conditions, such as the glucose rate of change. A feed-forward fully connected neural network was then proposed to exploit CGM information and some easily accessible patient parameters, with experimental evaluations on simulated data showing a small but statistically significant improvement in the blood glucose risk index. Simulated data is also used by Sun et al. in [9], where a basal-bolus advisor is trained using reinforcement learning in order to provide personalized suggestions to people with type 1 diabetes under multiple injections therapy.

The data-driven architecture proposed in this paper is generic in the sense that it can be trained to make recommendations about any variable that can impact BG levels, in particular carbohydrates and insulin. The task of making carbohydrate recommendations is potentially useful in scenarios where patients want to prevent hypoglycemia well in advance, or where a person is interested in achieving a relatively higher target BGL in preparation for an exercise event that is expected to lower it.

As a first step, in this paper we approach the problem of making carbohydrate recommendations. The rest of this paper is organized in the following way: Section 2 provides a more detailed description of the problem. Section 3 describes the model as well as the baselines used to compare against. Section 4 describes the dataset that is used and some of the features of the data. Section 5 discusses some of the training techniques and methods used as well as the results of the experiments that motivated the use of these techniques. Section 6 contains the conclusion and some plans for future work.

2 Three Carbohydrate Recommendation Scenarios

We assume that blood glucose levels are measured at 5 minute intervals through a CGM system. We also assume that discrete deliveries of insulin (boluses) and continuous infusions of insulin (basal rates)

[1] Ohio University, USA, email: {jb199113,bunescu,marling}@ohio.edu

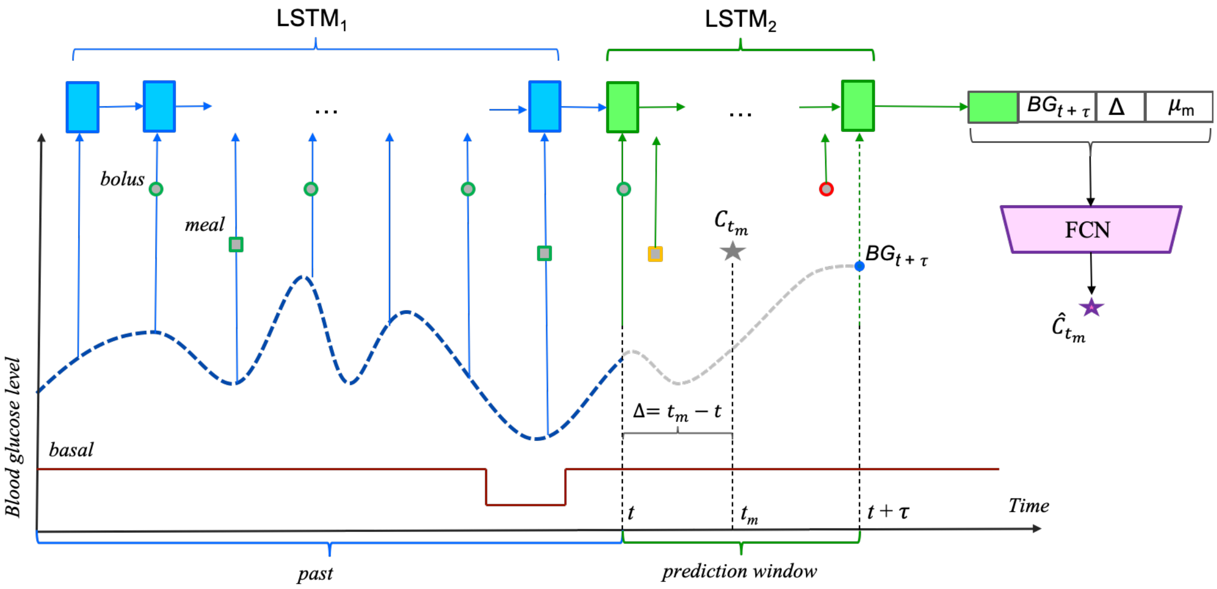

Figure 1. The general neural network architecture for carbohydrate recommendation. The dashed blue line in the graph represents a subject's BGL, while the solid brown line represents the basal rate of insulin. The gray star represents the meal at t_m. The other meals are represented by squares, and boluses are represented by circles. Meals and boluses with a green outline are allowed in all three example scenarios, while those with an orange outline are allowed in scenario S_2 and scenario S_3 examples, and those with a red outline are only allowed in scenario S_3 examples. The blue units in LSTM$_1$ receive input from different time steps in the past. The green units in LSTM$_2$ receive input from the prediction window. The purple trapezoid represents the 5 fully connected layers, whereas the output node at the end computes the carbohydrate prediction.

are recorded. Subjects provide the timing of meals and estimates of the amount of carbohydrates associated with each meal. Given the data available up to the present time t, the problem can formally be defined as predicting the number of grams of carbohydrates (number of carbs) C_{t_m} in a meal that is to be consumed at time $t_m \in [t, t+\tau)$ such that the person's BGL reaches a specified target value $BG_{t+\tau}$ at time $t+\tau$ in the future. Without loss of generality, in this paper we set the *prediction horizon* $\tau = 30$ and 60 minutes. We define three carbohydrate prediction scenarios, depending on whether events such as boluses or other meals happen inside the *prediction window* $[t, t+\tau)$:

1. **Scenario $\mathbf{S_1}$** assumes that there are no events in the prediction window $[t, t+\tau)$. Training a model for this scenario can be difficult due to the scarcity of corresponding training examples, as meals are typically preceded by boluses. The example shown in Figure 1 would be in this scenario if the orange and red outlined meals and boluses were not present.
2. **Scenario $\mathbf{S_2}$** subsumes scenario S_1 by allowing events before the meal, i.e. in the time window $[t, t_m]$. The example that is shown in Figure 1 would be a scenario S_2 example if the bolus outlined in red were not present, and would correspond to answering the following *what-if* question: how many carbs should be consumed at time t_m to achieve the target $BG_{t+\tau}$, if the meal were to be preceded by another meal and a bolus.
3. **Scenario $\mathbf{S_3}$** is the most general and allows events to happen during the entire prediction window $[t, t+\tau)$. The example in Figure 1 is a scenario S_3 example but not a scenario S_1 or scenario S_2 example because of the presence of the orange and red outlined meal and bolus.

We train and evaluate carbohydrate recommendation models for each scenario, using data acquired from 6 subjects with type 1 diabetes [5]. Given the scarcity of training examples for scenario S_1, our starting hypothesis is that models that are trained on examples from scenario S_3 will implicitly learn physiological patterns that will improve performance for the fewer examples in scenario S_1.

3 Baseline Models and Neural Architecture

Given training data containing meals with their corresponding timestamps and carbohydrates, we define the following baselines:

1. **Global average**: The average number of carbs over all of the meals in the subject's training data, μ, are computed and used as the estimate for all future meals, irrespective of context. This is a fairly simple baseline, as it predicts the same value for every example.
2. **ToD average**: In this Time-of-Day (ToD) dependent baseline, an average number of carbs is computed for each of the following five time windows during a day:
   - 12am-6am: μ_1 = early breakfast/late snacks.
   - 6am-10am: μ_2 = breakfast.
   - 10am-2pm: μ_3 = lunch.
   - 2pm-6pm: μ_4 = dinner.
   - 6pm-12am: μ_5 = late dinner/post-dinner snacks.

   The average for each ToD interval is calculated over all of the meals appearing in the corresponding time frame in the subject's training data. At test time, to predict the number of carbs for a meal to be consumed at time t_m, we first determine the ToD interval that contains t_m and output the corresponding ToD average.

Given sufficient historical data, the ToD baseline is expected to perform well for individuals who tend to eat very consistently and have regular diets. However, it is expected to perform poorly on individuals who have a lot of variation in their diets.

While simple to compute and use at test time, the two baselines are likely to give suboptimal performance, as their predictions ignore the history of BG values, insulin (boluses and basal rates), and meals, all of which could significantly modulate the effect a future meal might have on the BGL. To exploit this information, we propose the general neural network architecture shown in Figure 1. The first component in the architecture is a recurrent neural network (RNN) instantiated using Long Short-Term Memory (LSTM) cells [3], which is run over the previous 6 hours of data, up to the present time t. At each time step (5 minutes), this LSTM network takes as input the BGL, the carbohydrates, and the insulin dosages recorded at that time step. While sufficient for processing data corresponding to scenario S_1, this LSTM cannot be used to process events in the prediction window $[t, t+\tau)$ that may appear in scenarios S_2 and S_3, for which BGL values are not available. Therefore, in these scenarios, the final state computed by the first LSTM model (LSTM$_1$) at time t is projected and used as the initial state for a second LSTM model (LSTM$_2$) that is run over the time steps between $(t, t+\tau)$. The final state computed either by LSTM$_1$ (for scenario S_1) or LSTM$_2$ (for scenarios S_2 and S_3) is then used as input to a fully connected network (FCN) whose output node computes $\hat{C}_{t_m}$, an estimate of the carbohydrates at time t_m. Besides the LSTM final state, the input to the FCN contains the following additional features:

1. The target BGL at τ minutes into the future, i.e. $BG_{t+\tau}$.
2. The time interval $\Delta = t_m - t$ between the intended meal time and the present.
3. The ToD average computed for Baseline 2 corresponding to the time the meal was eaten.

The entire architecture is trained to minimize the mean squared error between the actual carbohydrates C_{t_m} recorded in the training data and the estimated value $\hat{C}_{t_m}$ computed by the output node of the FCN module. Each LSTM uses vectors of size 100 for the states and gates, whereas the FCN is built with 5 hidden layers, each consisting of 200 ReLU neurons, and one linear output node.

4 Dataset

The data used for the model was collected from 6 subjects with type 1 diabetes [5]. Information including the basal rate of insulin, boluses, meals, and BGL readings was collected over roughly 50 days, although the exact amount of time varies from subject to subject. This time series data is split into three sets, as follows: the last 10 days of data for each subject are used as testing, the previous 10 days are used as validation, and the remainder of the data is used for training.

4.1 From Meal Events to Examples

Since the total number of available examples is directly related to the number of meals, it is useful to know how many meals each subject had. This is shown in Table 1, together with the average number of carbs per meal (Avg), and the corresponding standard deviation (StdDev). Most subjects have a similar average number of carbohydrates in their meals, with the exception of 570 who has a significantly larger number of carbs per meal on average, and more importantly, a much higher standard deviation than the other subjects.

A meal event occurring at time t_m may give rise to multiple examples, depending on the position of t_m in the interval $[t, t+\tau)$. When $\tau = 30$ minutes, an example is created for every possible position of t_m within $[t, t+\tau)$. However, when $\tau = 60$ minutes, an example is created for every position of t_m within $[t, t+30]$, to ensure that there are at least 30 minutes between the meal and the prediction horizon. Table 2 below shows the resulting number of examples for $\tau = 30$ and 60 minutes, in each of the three scenarios. Note that there are fewer examples in scenarios S_1 and S_2 when $\tau = 60$ vs. 30 minutes, despite there being more scenario S_3 examples. This can be explained by the scenarios S_1 and S_2 criteria being even more difficult to meet when $\tau = 60$ minutes, i.e. there cannot be any event within $[t, t+60)$ for S_1, or any event within $[t_m, t+60)$ for S_2.

Table 1. Meal statistics, per subject and total.

		Carbs Per Meal	
Subject	Meals	Avg	StdDev
559	179	36.0	16.0
563	153	29.9	16.3
570	169	105.3	42.0
575	284	40.6	22.9
588	257	30.8	16.6
591	249	31.6	14.2
Total	1291	43.5	33.1

Table 2. Example counts by scenario, for 30 and 60 minutes.

	Scenario S_1		Scenario S_2		Scenario S_3	
Dataset	30	60	30	60	30	60
Training	2396	1923	3889	3491	5096	5931
Validation	629	510	1061	981	1388	1626
Testing	469	339	950	851	1236	1435
Total	3494	2772	5900	5323	7720	8992

5 Experimental Evaluation

The Adam [4] variant of gradient descent is used for training, with the learning rate and mini-batch size being tuned on the validation data. In an effort to avoid overfitting, early stopping with a patience of 5 epochs and dropout with a rate of 10% are used for both models. Interestingly, dropout was found to help the model if it was only applied to the LSTM networks of the model at each time step and not the fully connected network.

Since the overall number of examples available in the dataset is low, the performance was improved by first pretraining a generic model on the combined data from all 6 subjects. Then, for each subject, a new model is initialized with the weights of the generic model, and then fine-tuned on the subject's training data. For each subject, five models were trained with different seedings of the random number generators. We also experimented with fine-tuning models on the union of the training and validation data instead of just the training data. When this combined data is used, the average carb values used in the baselines are recalculated over the union of the training and validation data for each subject.

5.1 Results

The metrics used to evaluate the performance of the models are the root mean squared error (RMSE) and the mean absolute error (MAE), which is less sensitive to large errors. At the end of the training process, there are five fine-tuned models for each subject. The *average* RMSE and MAE of the five models are reported, as well as the RMSE and MAE of the *best* model. The model that is considered the ”best” is the one that had the lowest MAE on the validation data. The results of the five models for each subject are also averaged across all subjects to obtain one overall RMSE and one overall MAE value for the *average model* and the *best model* scores. The baselines are treated much the same, as their RMSE and MAE values are averaged across all subjects to give an RMSE and an MAE score for each baseline.

Table 3 compares the validation results achieved in scenario S_3 by models with and without pretraining for $\tau = 30$ minutes. This experiment clearly shows the benefit of pretraining the models: both the RMSE and MAE are noticeably lower for the pretrained models. As a result, pretraining is always used as part of the training process for both values of τ.

Table 3. Results with and without pretraining, $\tau = 30$.

Setting	RMSE	MAE
Without Pretraining	22.2	15.5
With Pretraining	**20.7**	**14.5**

Table 4 compares models that were fine-tuned on training and validation data with models fine-tuned solely on the training data, in scenario S_3. The results show that the extra examples provided by the validation data proved helpful in improving performance. It is interesting to note that using the combined training-validation data only slightly helped the baselines, but helped the LSTM-based models by a noticeable margin.

Table 4. Fine-tuning on Training vs. Training ∪ Validation, $\tau = 30$.

Fine-tuning	Baselines & Models	RMSE	MAE
Training	Global Average	23.3	19.2
	ToD Average	22.5	17.8
	Average Model	21.3	16.0
	Best Model	**20.7**	**15.3**
Training ∪ Validation	Global Average	23.1	19.0
	ToD Average	22.2	17.7
	Average Model	20.1	15.0
	Best Model	**19.2**	**14.2**

Table 5 compares the Baselines (Global and ToD averages) with the trained Models (Best and Average) in terms of their RMSE and MAE in the three scenarios.

Table 5. Results for scenarios S_1, S_2, and S_3, for $\tau = 30$ and 60 minutes.

		RMSE		MAE	
	Baselines & Models	30	60	30	60
S_1	Global Average	19.7	18.4	15.7	15.0
	ToD Average	**18.9**	**17.6**	14.8	14.4
	Average Model	19.3	19.5	14.1	**13.9**
	Best Model	19.0	19.8	**13.9**	**13.9**
S_2	Global Average	18.4	17.1	14.5	13.8
	ToD Average	17.4	15.9	13.1	12.2
	Average Model	16.2	**15.3**	11.9	11.4
	Best Model	**15.8**	15.4	**11.6**	**10.9**
S_3	Global Average	18.5	18.6	14.6	14.7
	ToD Average	17.5	17.6	13.2	13.3
	Average Model	15.7	15.6	11.5	11.3
	Best Model	**15.6**	**14.8**	**11.4**	**10.6**

Overall, the LSTM-based models (Average or Best) had the best RMSE and MAE performance across all three scenarios, with the exception of the RMSE scores for scenario S_1. Compared to the other two scenarios, the LSTM models and the baselines have a lower performance in S_1. The decline in performance is even more apparent for the LSTM models, which cannot beat the time-dependent baseline in terms of RMSE for both the 30 minute and 60 minute prediction horizons. This can be explained by the limited number of examples for scenario S_1: since there are so few testing examples in this scenario per subject, one bad prediction can hurt the results significantly, more so for the RMSE than the MAE. Furthermore, the trained models tend to make very similar predictions for all examples stemming from a specific meal, meaning that if the model made a bad prediction for one test example, it likely made a series of similarly bad predictions.

To alleviate the scarcity of training examples in scenario S_1, models trained on S_3 examples, which are the most plentiful and subsume S_1, were evaluated separately on test examples from S_1. This gives an indication on whether any transfer learning is taking place. Table 6 shows the results of this transfer learning experiment, indicating that training on the additional examples from scenario S_3 helps improve performance on scenario S_1 to the level that now the LSTM-based models outperform both baselines.

Table 6. Comparative performance on scenario S_1 test examples: Baselines vs. LSTM-based models trained on S_1 and S_3 examples.

		RMSE on S_1		MAE on S_1	
	Baselines & Models	30	60	30	60
	Global Average	19.7	18.4	15.7	15.0
	ToD Average	18.9	17.6	14.8	14.4
Training on S_1	Average Model	19.3	19.5	14.1	13.9
	Best Model	19.0	19.8	13.9	13.9
Training on S_3	Average Model	**18.2**	17.6	**13.6**	13.3
	Best Model	18.3	**16.7**	13.8	**13.0**

6 Conclusion and Future Work

We introduced a generic neural architecture, composed of two chained LSTMs and a fully connected network, with the purpose of training data-driven models for making recommendations with respect to any type of quantitative events that may impact BG levels, in particular carbohydrate amounts and bolus insulin dosages. Experimental evaluations on the task of carbohydrate recommendations within a 30 or 60 minute prediction window demonstrate the feasibility and potential of the proposed architecture, as well as its ability to benefit from pre-training and transfer learning. Future plans include evaluating carbohydrate recommendations within larger prediction windows, as well as training the architecture for bolus recommendations.

ACKNOWLEDGEMENTS

This work was supported by grant 1R21EB022356 from the National Institutes of Health (NIH). Conversations with Josep Vehi helped shape the research directions presented herein. The contributions of physician collaborators Frank Schwartz, MD, and Amber Healy, DO, are gratefully acknowledged. We would also like to thank the anonymous people with type 1 diabetes who provided their blood glucose, insulin, and meal data.

REFERENCES

[1] R. Bunescu, N. Struble, C. Marling, J. Shubrook, and F. Schwartz, 'Blood glucose level prediction using physiological models and support vector regression', in *Proceedings of the Twelfth International Conference on Machine Learning and Applications (ICMLA)*, pp. 135–140. IEEE Press, (2013).

[2] G. Cappon, M. Vettoretti, F. Marturano, A. Facchinetti, and G. Sparacino, 'A neural-network-based approach to personalize insulin bolus calculation using continuous glucose monitoring', *Journal of Diabetes Science and Technology*, **12**(2), 265–272, (2018).

[3] Sepp Hochreiter and Jürgen Schmidhuber, 'Long short-term memory', *Neural Computation*, **9**, 1735–1780, (12 1997).

[4] D. P. Kingma and J. L. Ba, 'Adam: A method for stochastic optimization', in *Third International Conference for Learning Representations (ICLR)*, San Diego, California, (2015).

[5] C. Marling and R. Bunescu, 'The OhioT1DM dataset for blood glucose level prediction', in *The 3rd International Workshop on Knowledge Discovery in Healthcare Data*, Stockholm, Sweden, (2018). Available at http://ceur-ws.org/Vol-2148/paper09.pdf.

[6] S. Mirshekarian, H. Shen, R. Bunescu, and C. Marling, 'LSTMs and neural attention models for blood glucose prediction: Comparative experiments on real and synthetic data', in *Proceedings of the 41st International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2019)*, Berlin, Germany, (2019).

[7] S. G. Mougiakakou and K. S. Nikita, 'A neural network approach for insulin regime and dose adjustment in type 1 diabetes', *Diabetes Technology & Therapeutics*, **2**(3), 381–389, (2000).

[8] K. Plis, R. Bunescu, C. Marling, J. Shubrook, and F. Schwartz, 'A machine learning approach to predicting blood glucose levels for diabetes management', in *Modern Artificial Intelligence for Health Analytics: Papers Presented at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 35–39. AAAI Press, (2014).

[9] Q. Sun, M. V. Jankovic, J. Budzinski, B. Moore, P. Diem, C. Stettler, and S. G. Mougiakakou, 'A dual mode adaptive basal-bolus advisor based on reinforcement learning', *IEEE Journal of Biomedical and Health Informatics*, **23**(6), 2633–2641, (2019).

[10] J. Walsh, R. Roberts, T. S. Bailey, and L. Heinemann, 'Bolus advisors: Sources of error, targets for improvement', *Journal of Diabetes Science and Technology*, **12**(1), 190–198, (2018).

[11] H. C. Zisser, L. T. Robinson, W. Bevier, E. Dassau, C. L. Ellingsen, F. J. Doyle, and L. Jovanovic, 'Bolus calculator: A review of four "smart" insulin pumps.', *Diabetes Technology & Therapeutics*, **10**(6), 441–444, (2008).