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

Jeremy Beauchamp and Razvan Bunescu and Cindy Marling

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 BGL 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)
Scenario 1 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 time-stamps 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, \(\mu\), 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: \(\mu_1\) = early breakfast/late snacks.
   - 6am-10am: \(\mu_2\) = breakfast.
   - 10am-2pm: \(\mu_3\) = lunch.
   - 2pm-6pm: \(\mu_4\) = dinner.
   - 6pm-12am: \(\mu_5\) = 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 those 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 \( \hat{C}_{tm} \), 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 \( \tau \) 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 \( C_{tm} \) recorded in the training data and the estimated value \( \hat{C}_{tm} \) 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 (Std-Dev). 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 \([t, t + 60]\), 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 \( S_2 \).

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 $\tau$.

<table>
<thead>
<tr>
<th>Setting</th>
<th>RMSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Pretraining</td>
<td>22.2</td>
<td>15.5</td>
</tr>
<tr>
<td>With Pretraining</td>
<td>20.7</td>
<td>14.5</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Fine-tuning</th>
<th>Baselines &amp; Models</th>
<th>RMSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>Global Average</td>
<td>23.3</td>
<td>19.2</td>
</tr>
<tr>
<td></td>
<td>ToD Average</td>
<td>22.5</td>
<td>17.8</td>
</tr>
<tr>
<td></td>
<td>Average Model</td>
<td>21.3</td>
<td>16.0</td>
</tr>
<tr>
<td></td>
<td>Best Model</td>
<td>20.7</td>
<td>15.3</td>
</tr>
<tr>
<td>Training ∪ Validation</td>
<td>Global Average</td>
<td>23.1</td>
<td>19.0</td>
</tr>
<tr>
<td></td>
<td>ToD Average</td>
<td>22.2</td>
<td>17.7</td>
</tr>
<tr>
<td></td>
<td>Average Model</td>
<td>20.1</td>
<td>15.0</td>
</tr>
<tr>
<td></td>
<td>Best Model</td>
<td>19.2</td>
<td>14.2</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Baselines &amp; Models</th>
<th>RMSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Average</td>
<td>19.7</td>
<td>18.8</td>
</tr>
<tr>
<td>ToD Average</td>
<td>18.9</td>
<td>17.6</td>
</tr>
<tr>
<td>Average Model</td>
<td>19.3</td>
<td>17.5</td>
</tr>
<tr>
<td>Best Model</td>
<td>19.0</td>
<td>17.4</td>
</tr>
</tbody>
</table>

Table 5. Results with and without pretraining, $\tau = 30$.

Table 6. Comparative performance on scenario $S_1$ test examples: Baselines vs. LSTM-based models trained on $S_1$ and $S_2$ examples.

<table>
<thead>
<tr>
<th>Baselines &amp; Models</th>
<th>RMSE on $S_1$</th>
<th>MAE on $S_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Average</td>
<td>19.7</td>
<td>18.4</td>
</tr>
<tr>
<td>ToD Average</td>
<td>18.9</td>
<td>17.6</td>
</tr>
<tr>
<td>Average Model</td>
<td>19.3</td>
<td>19.5</td>
</tr>
<tr>
<td>Best Model</td>
<td>19.0</td>
<td>19.8</td>
</tr>
</tbody>
</table>

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


