Appendix 1. Model building and training in the recurrent neural network

Recurrent neural network long short-term memory (RNN-LSTM) was developed to solve long-range dependency and vanishing gradient problems seen in RNN, which degraded performance as event length increased. Our proposed LSTM model is designed with the following structure. Our proposed LSTM model is designed with the following structure. For the optimization of the algorithm, RMSProp [31] was used to update parameters through back-propagation. The learning sample set S is consisted of ordered pairs (x, y) with inputs and correct answers. Input x is an element of input set X, whereas correct answer y is an element of correct answers set Y. Here, x is consisted of serial data in the form of an individual's information. The T in input $x=(x_1,\ldots, x_t, \ldots, x_T)$ represents the length of a sample (i.e., the number of events) and varies individually. A health examination record is consisted of an event; and, all events $x_1, \ldots, x_T$ for a person are in chronological order. x_t represents an event (examination record) at a specific time and is used as a vector with features. The correct answer $y=(y_1, \ldots, y_K)$ has a Boolean value indicating whether or not type 2 diabetes mellitus (T2DM) occurred in the past. Hyper-parameters at a learning rate of 0.01 were configured, a dropout probability of 50%, and a mini-batch of 64. The correct answer is one-hot encoded to be used as cross entropy in a loss function. K—the number of classes—is set as 2. Assuming the output of prediction model is $\hat{y}=(\hat{y}_1, \ldots, \hat{y}_K)$, cross entropy as shown in Equation 1 is used for our loss function.

$$E(y,\hat{y}) = -\sum_{i=1}^{K}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad \text{(Equation 1)}$$

The output sequence for LSTM was $o=(o_1, \ldots o_t, \ldots, o_T)$ where T was the event length being the same as the event length of x. The prediction result of an input sample provided the probability of T2DM in the near future taken the occurrence of past event into account.

Only the last output o_T among $o_1, \ldots o_t, \ldots, o_T$ was used and reflected to an output z as shown in Equation 2. Here, $W \in R^{K \times H}$ was the parameter to be optimized and H was the number of hidden nodes in the last hidden layer. To calculate the probability of $\hat{y}_i$ from z, the softmax function was used as shown in Equation 3.

$$z = W\mathbf{o}_T \quad \text{(Equation 2)}$$

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad \text{for } i = 1, \ldots, K \quad \text{(Equation 3)}$$

LSTM had a memory cell with input, forget and output gates. Each LSTM unit uses the equations in Equation 4 which are commonly used in LSTM.

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \\ h_t &= o_t \circ \tanh(c_t) \end{aligned} \quad \text{(Equation 4)}$$

σ was the logistic sigmoid function and i, f, o, and c were respectively the input gate, forget gate, output gate, and cell. The subscripts in weight matrix above have an obvious meaning. For example, W_{hi} is the hidden-input gate matrix while W*xo* being the input-output gate matrix. The bs are bias terms which are added for i, f, o, and c equations. Let N be the number of LSTM blocks and M the number of inputs, W_i, W_f, W_c, $W_o \in R^{N \times M}$.