Final Project:

It involves cumulative activities of NN. Your Final project is to use an application/experiments on (preferred novel) public dataset (i.e., not used in the referenced papers listed on the class website) with modification of one of the "basic" Gated RNN (LSTM, GRU, MGU RRNs). You may (or may not) choose to include/integrate ConvNet for (classification) depending on your experiments. Your model-type (LSTM, GRU, or MGU) will be assigned to you. (If you desire a different Gated RNN type, Plz let me know via email ASAP). You may choose your own code. However, you will also be provided with KERAS inherence class for several modifications which you may use or alter.

The key elements of your project are:
1) The choice of the (novel) dataset(s),
2) The modification from the set of provided modifications, which you will employ to metrically compare its performance with the (assigned) standard Gated RNN
3) experiments/simulations, compared with the literature
4) Report formatted in the form of a Conference paper (e.g., IEEE conf template, a sample will be provided). Your report should take the professional form of a paper that could potentially be published on http://arXiv.org.

Define your experiments and comparisons on the same dataset and experiments and summarize your findings. The goal is to have your paper potentially publishable on the arXiv website.

Model0: If one considers the standard LSTM model→ LSTM0

It has 3 Gates:

$$i_t = \sigma(U_i h_{t-1} + W_i x_t + b_i)$$
$$f_t = \sigma(U_f h_{t-1} + W_f x_t + b_f)$$
$$o_t = \sigma(U_o h_{t-1} + W_o x_t + b_o)$$

Memory Cells & hidden units:

$$c_t = f_t * c_{t-1} + i_t * \tanh(U h_{t-1} + W x_t + b)$$
$$h_t = o_t * \tanh(c_t)$$

And the output layer

$$y_t = V h_t + d$$

With a possible additional softmax (layer) to represent the output as confidence measure or a probability measure (i.e., $p_t = soft\max(y_t)$)

(Note: * is used as in Python, for a point-wise (Hadamard) multiplication.)

The Modifications:

The modifications are applied to all Gates uniformly for modularity: (we show the modifications here applied to one gate, say, i)

0) The full gate signals (Model0: LSTM0)

$$i_t = \sigma(U_i h_{t-1} + W_i x_t + b_i)$$

Is driven by 3 variable. Thus, there are 6 other possible variations, 3 without the external input signal x_t, and 3 with the input signal. For efficiency, we consider the 3 without the external input signal as the input over time is contained in the hidden unit.

In your final project, you will compare, in metric performance the standard model (e.g, LSTM, GRU, MGU, assigned to you) with one or more modifications in the gate(s) (again, assigned to you). (IF you would like a different model, please let me know asap via, Subject ECE885).

For one of the gates, say i in this case, one has

Case 1: Model1→ LSTM1

$i_t = \sigma(U_i h_{t-1} + b_i)$, No input signal

Case 2: Model2 → LSTM2

$i_t = \sigma(U_i h_{t-1})$ No input signal, no bias

Case 3: Model 3 → LSTM3

$i_t = \sigma(b_i)$ No input signal, no hidden unit signal.

Similarly, one can write the Models: GRU, GRU1-GRU3, and MGU, MGU1-3.

Your report paper should include the description and itemized summary of your results highlighting the key findings in sections. Suggested sample sections include:

1) Introduction: Purpose of the project
2) Literature review: starting from the references on D2L and what they have accomplished. Provide metrics/numbers when summarizing their performances
3) The Architecture: the detailed Network (Architecture) description with hyper-parameters, optimizers, etc.
4) Database application: Your choice of novel public database. (See reference papers and similar references, Libraries, etc.)

5) Conclusions and Lessons learned: what worked and failed. What new things have you learned/discovered if any from this application experiment(s) that are better than what is reported in the literature. What you or another investigator may do in the future based on your experience in this project.

Assignment:

I) Model assignment for the Final Project are below. Your model code is given next to 1-3 letters of your last name. Two other models may be added later.

Using Model 1:LSTM
LSTM1: Last names begin with: A, Ba
LSTM2: Last names begin with: Be, Do
LSTM3: Last names begin with: Ho, Hu

Using Model 2: GRU

GRU1: Last names begin with: La
GRU2: Last names begin with: Li, N
GRU3: Last names begin with: Sh, Su

Using Model 3: MGU

MGU1: Last names begin with: Wa
MGU2: Last names begin with: We, Y
MGU3: Last names begin with: Z

Supp Material:
Remark: If you will be using **Keras** to implement your model in your assignment(s), you may place the new module (posted on the assignment folder):

recurrent.py
in your Conda or miniconda sub-folder (under the folder keras\layers)
..Miniconda\Lib\site-packages\keras\layers
Now you should have classes/Layers called:
LSTM (for Model 1)
GRU (for Model 2)
And
GRU1 (for Model 3)

I will also upload Keras inheritance classes for the models which you can use or alter.

Three Models of Gated RNNs

Consider the 3 basic RNN models with gating:

Model 1: Consider the Basic LSTM model
With 3 Gates:

$$i_t = \sigma(U_i h_{t-1} + W_i x_t + b_i)$$
$$f_t = \sigma(U_f h_{t-1} + W_f x_t + b_f)$$
$$o_t = \sigma(U_o h_{t-1} + W_o x_t + b_o)$$

Memory Cells & hidden units:

$$c_t = f_t * c_{t-1} + i_t * \tanh(Uh_{t-1} + Wx_t + b)$$
$$h_t = o_t * \tanh(c_t)$$

And the output layer

$$y_t = Vh_t + d$$

With a possible additional softmax (layer) to represent the output as confidence measure or a probability measure (i.e., $p_t = soft\max(y_t)$)

Model 2: Consider the Basic GRU model

With 2 Gates:

$$i_t = \sigma(U_i c_{t-1} + W_i x_t + b_i)$$
$$o_t = \sigma(U_o c_{t-1} + W_o x_t + b_o)$$

Memory Cells & hidden units:

$$c_t = (1 - i_t) * c_{t-1} + i_t * \tanh(Uh_{t-1} + Wx_t + b)$$
$$h_t = o_t * c_t$$

And the output layer

$$y_t = Vh_t + d$$

With a possible additional softmax (layer) to represent the output as confidence measure or a probability measure (i.e., $p_t = soft\max(y_t)$)

Model 3: Consider the Minimal GRU model

With one Gate equation:

$$i_t = \sigma(U_i c_{t-1} + W_i x_t + b_i)$$

Memory Cells & hidden units:

$$c_t = (1 - i_t) * c_{t-1} + i_t * \tanh(Uh_{t-1} + Wx_t + b)$$
$$h_t = i_t * c_t$$

And the output layer

$$y_t = Vh_t + d$$

With a possible additional softmax (layer) to represent the output as confidence measure or a probability measure (i.e., $p_t = soft\max(y_t)$)

References:

The references/Papers, and the references therein, related to these 3 models, are posted on D2L. They include experiments/comparison on datasets. You are to use, e.g., the MNIST dataset to test your model for 100 hidden/cell units. See more details on selecting the hyper parameters and epochs in the plots in the references.