Chapter20-1 Recurrent Neural Network

1. Slot Filling
   a) 1-of-N Encoding
   b) Beyond 1-of-N Encoding
      i) Dimension of “other”
      ii) Word hashing
2. SimpleRNN
   a) The output of hidden layer are stored in the memory
   b) Memory can be considered as another input
   c) Changing the sequence order will change the output
   d) Elman RNN
      i) Hidden layer → hidden layer
   e) Jordan RNN
      i) Output layer → hidden layer
   f) Bidirectional RNN
3. Long Short-term Memory
   a) Four Inputs: Input, input gate signal, output gate signal, forget gate signal
   b) One Cell: Memory Cell
   c) One Output: Output of output gate
   d) The activation function of gates usually is sigmoid function
   e) Usually 4 times of parameters than other neural networks
4. Optimize RNN
   a) Back Propagation Through Time (BPTT)
   b) RNN-based network is not always easy to learn
      i) The error surface is tough
      ii) Surface is either very flat or very steep
      iii) Clipping (if gradient > threshold => gradient = threshold)
   c) The reason is that weight of memory to neural is used repeatedly over time
5. Helpful Techniques
   a) LSTM
      i) Deal with the problem of gradient vanishing (take flat places off)
      ii) Can’t deal with the problem of gradient explode
      iii) Input are added into memory, not format memory in RNN
      iv) The influence never disappears unless forget gate is closed
   b) GRU
      i) LSTM has 3 gates, whereas GRU only has 2 gates
      ii) Spirit: Old gone, new come
      iii) When the input gate is opened, the forget gate is automatically closed
      iv) Need to clear the value in the memory to put the new value in
   c) Clockwise RNN
   d) Structurally Constrained Recurrent Network (SCRN)
   e) Hinton’s Trick:
      Vanilla RNN initialized with Identity Matrix + ReLU
      Outperform or be comparable with LSTM

6. RNN v.s. Structured Learning
    a) RNN, LSTM
        i) Unidirectional RNN does not consider the whole sequence
        ii) Cost and error not always related
        iii) Deep
    b) HMM, CRF, Structured Perception/SVM
        i) Using Viterbi, so consider the whole sequence
        ii) Explicitly consider the label dependency
        iii) Cost is the upper bound of error
    c) Integrate together : Deep Learning + Structured Learning
        i) Speech Recognition : CNN/LSTM/DNN + HMM
        ii) Semantic Tagging : Bi-directional LSTM + CRF/Structured SVM
    d) GAN
        i) Problem 1 : Evaluation Function $F(x)$ ⇔ Discriminator
        ii) Problem 2 : Inference : x = argmax $F(x)$ ⇔ Generator
        iii) Problem 3 : You know how to learn $F(x)$
        iv) Conditional GAN