

MD-LSTM

Goal

Applied RNN to multi-dimensional data rather than sequence.

Approach

Recap standard LSTM first

$$\mathbf{f}_t = \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f) \quad (1)$$

$$\mathbf{i}_t = \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i) \quad (2)$$

$$\mathbf{o}_t = \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o) \quad (3)$$

$$\hat{\mathbf{c}}_t = \tanh(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c) \quad (4)$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \hat{\mathbf{c}}_t \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t) \quad (6)$$

-
1. Replace the single recurrent connection found in standard RNNs with *as many recurrent connections* as there are dimensions in the data.
 2. During the forward pass, at each point in the data sequence, the hidden layer of the network receives both an external input and its activations from one step back along all dimensions.

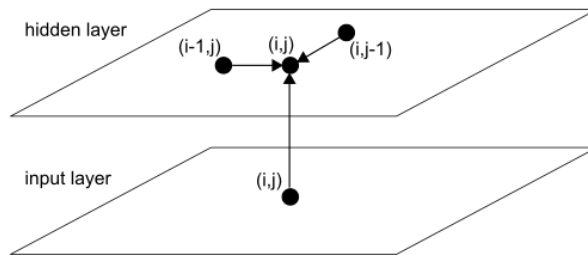


Figure 1: 2D RNN Forward pass.

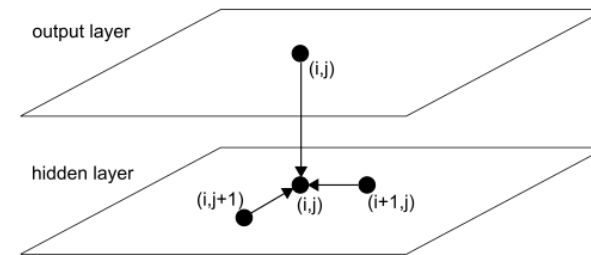


Figure 2: 2D RNN Backward pass.

Fig1. Figure 1 and 2 from the paper.

Quoted from the paper:

*the data must be processed in such a way that when the network reaches a point in an n -dimensional **sequence**, it has already passed through all the points from which it will receive its **previous** activations.*

3. Incorporate directional information (See Fig2.).

- Bi-direction contains two separate hidden layers that process the input sequence in the forward and reverse directions. – The two hidden layers are connected to a single output layer to provide the network both past and future context.

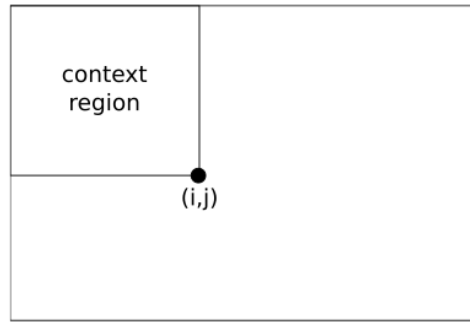


Figure 4: Context available at (i,j) to a 2D RNN with a single hidden layer.

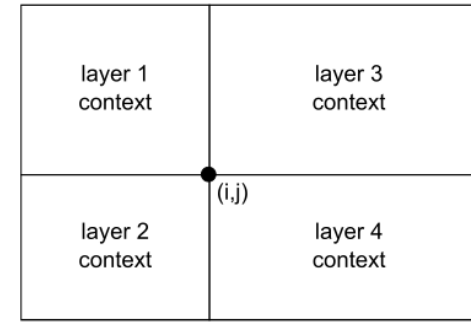


Figure 5: Context available at (i,j) to a multi-directional 2D RNN.

Fig2. Figure 4 and 5 from the paper.

Inputs

1. N : the dimension number
2. Input \mathbf{x} that is arranged in an N -dimensional grid
 - such as 2-D grid of pixels in an image.
3. N hidden vectors $\mathbf{h}_1, \dots, \mathbf{h}_N$
4. N memory vectors $\mathbf{c}_1, \dots, \mathbf{c}_N$

$$\mathbf{c}_t = \sum_i^N \mathbf{f}_t^i \circ \mathbf{c}_t^i + \mathbf{i}_t \circ \hat{\mathbf{c}}_t$$

Quoted from the paper:

The standard formulation of LSTM is explicitly one-dimensional, since the cell contains a single self connection, whose activation is controlled by a single forget gate. However we can easily extend this to n dimensions by using instead n self connections (one for each of the cell's previous states along every dimension) with n forget gates.

My Note:

1. Each direction calculates its own forget gate. For example, the input data is arranged as a N dimensional grid, then, N forget gate is calculated.
2. All inputs in the t -th iteration calculate a single input, output gate, and a single cell candidate.

References

1. [Multi-Dimensional Recurrent Neural Networks](#)

2. [A TensorFlow implemented Multi Dimensional Recurrent Networks on github](#)
3. [A stackoverflow question on MD-LSTM in TF](#)