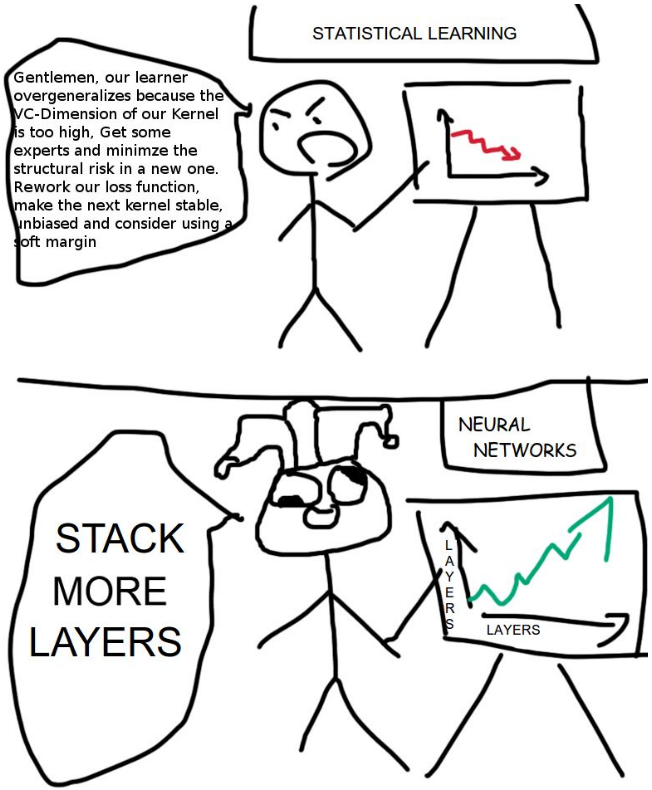
GRID-LSTM

MAS.S63, Eric Chu


STATISTICAL LEARNING
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high, Get some experts and minimze the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin
NEURAL NETWORKS
STACK MORE LAYERS
LAYERS
LAYERS

Neural Networks++

Memory

Attention

Bayesian

Extremely deep

A LSTM Recap

Motivation: problems with vanilla RNNs

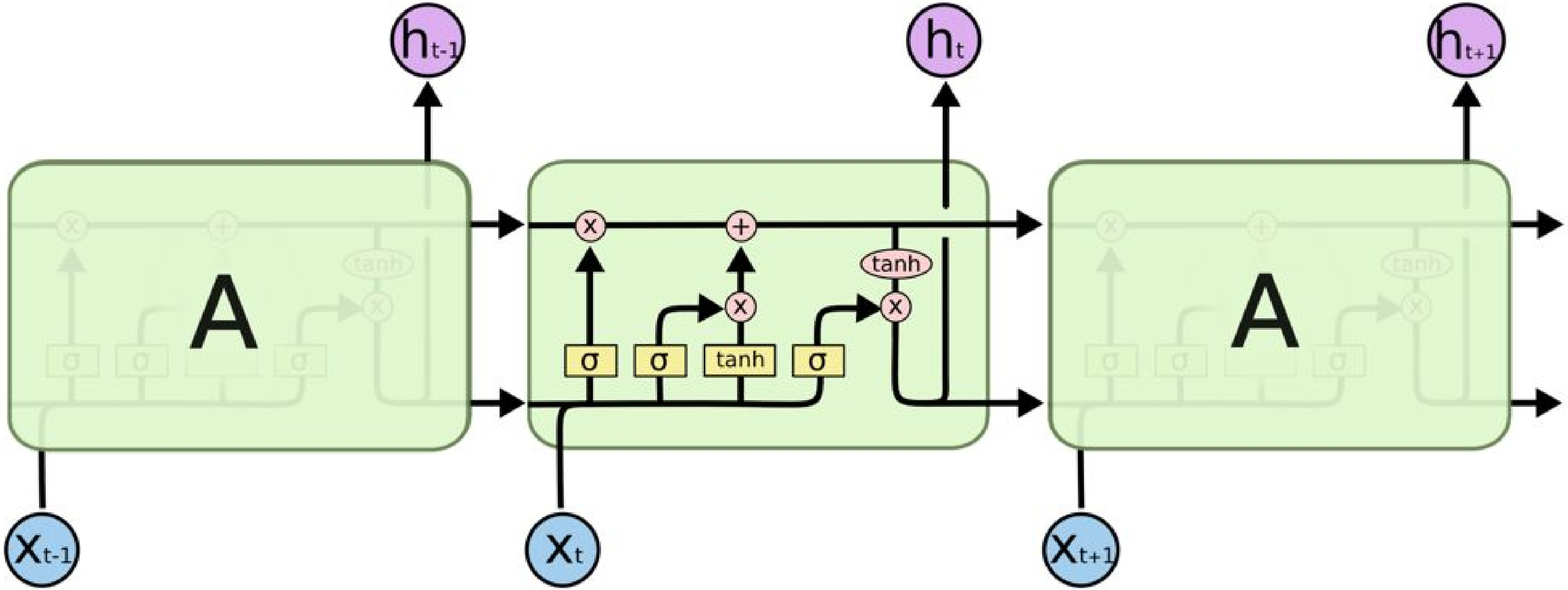
Vanishing gradient due to non-linearities

Harder to capture longer term interactions

Solution:

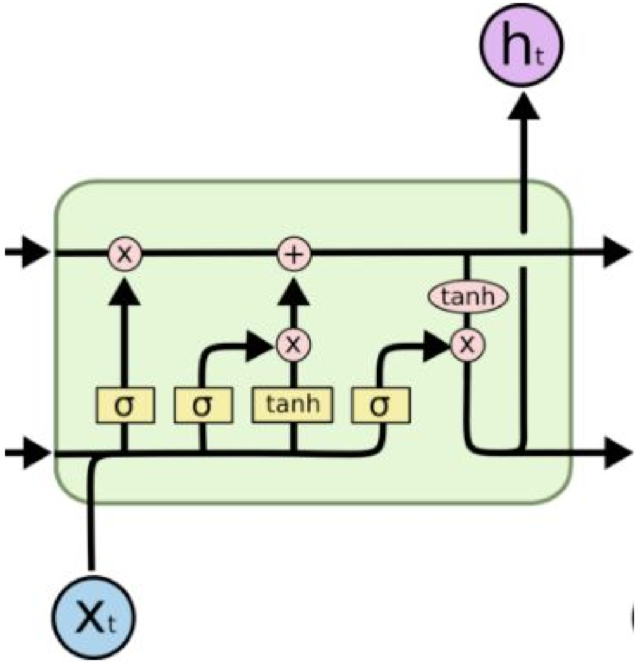
“Long” “short-term” memory cells, controlled by gates that allow information to pass unmodified over many timesteps

A LSTM Recap



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

A LSTM Recap



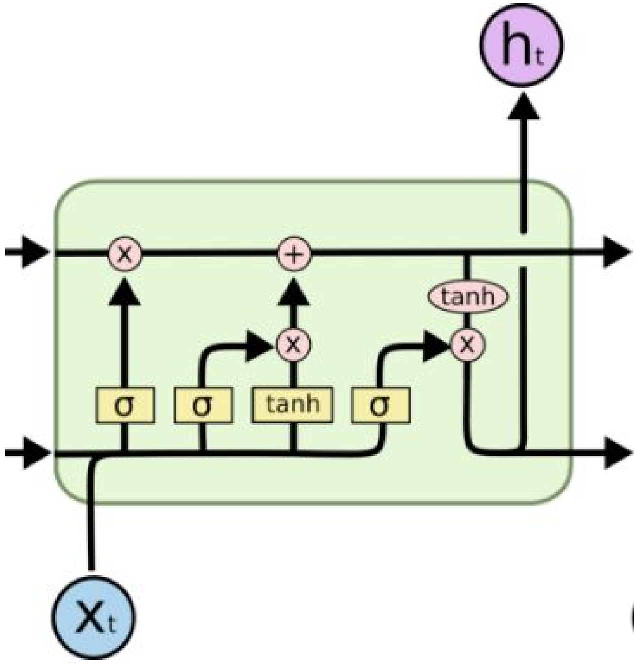
Forget gate: which parts of memory vector to delete

Input gate: which parts of memory vector to update

Content gate: what should the memory vector be updated with

Output gate: what gets read from new memory into hidden vector

A LSTM Recap



$$i_t = g(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

$$f_t = g(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

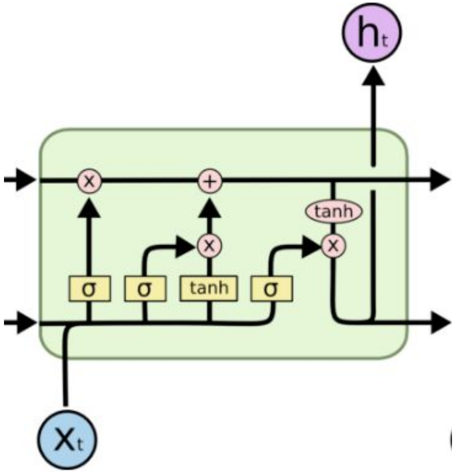
$$o_t = g(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

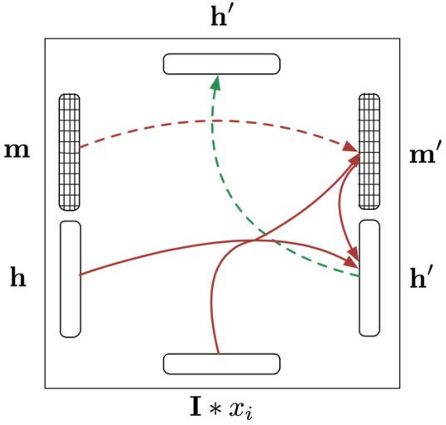
$$c\_in_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_{c\_in})$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c\_in_t$$

$$h_t = o_t \cdot tanh(c_t)$$

A LSTM Recap



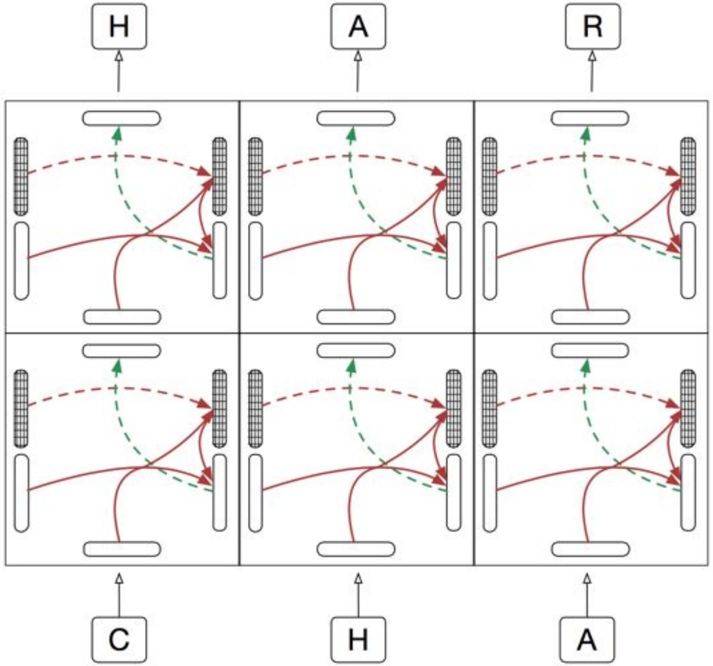


Standard LSTM block

$$\mathbf{H} = \begin{bmatrix} I x_i \\ \mathbf{h} \end{bmatrix}$$

$$\mathbf{g}^u = \sigma(\mathbf{W}^u \mathbf{H})$$
$$\mathbf{g}^f = \sigma(\mathbf{W}^f \mathbf{H})$$
$$\mathbf{g}^o = \sigma(\mathbf{W}^o \mathbf{H})$$
$$\mathbf{g}^c = \tanh(\mathbf{W}^c \mathbf{H})$$
$$\mathbf{m}' = \mathbf{g}^f \odot \mathbf{m} + \mathbf{g}^u \odot \mathbf{g}^c$$
$$\mathbf{h}' = \tanh(\mathbf{g}^o \odot \mathbf{m}')$$

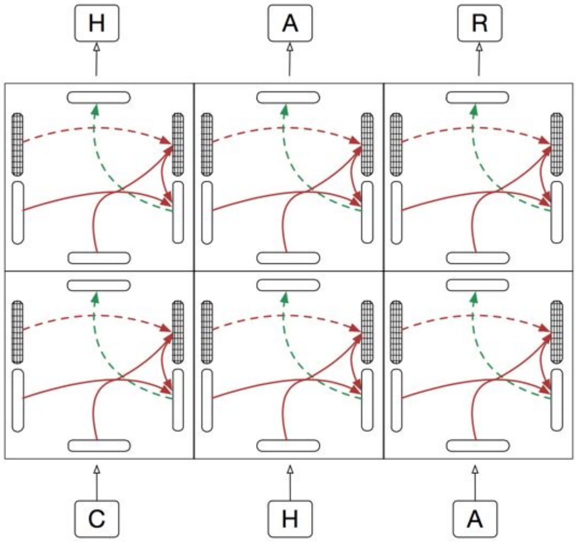
A LSTM Recap: Stacked LSTM



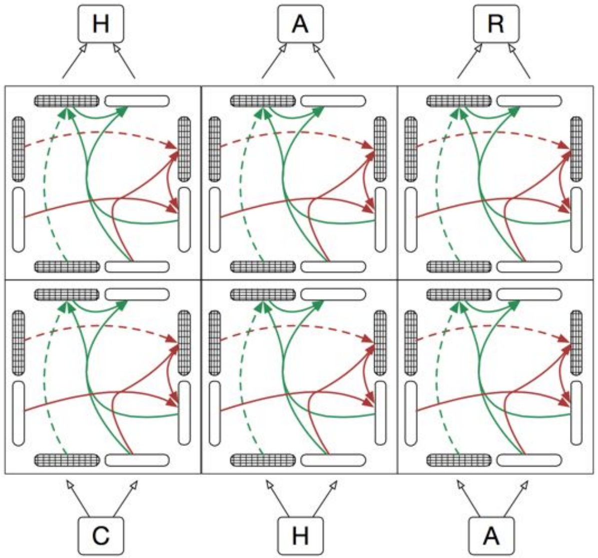
Stacked LSTM

Grid-LSTM: Motivation

Stacked LSTM, but LSTM units connections along depth dimension as well as temporal dimension

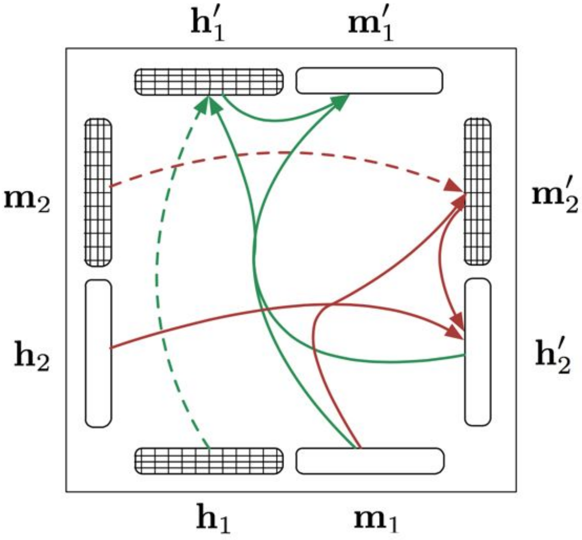


Stacked LSTM



2d Grid LSTM

Grid-LSTM: Motivation

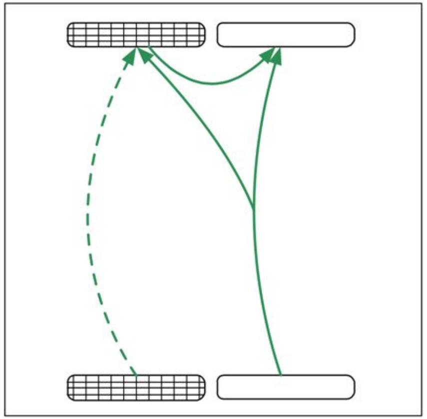


2d Grid LSTM block

Grid-LSTM: 1D

1D Grid-LSTM = feedforward NN with LSTM cells instead of transfer functions such as tanh and ReLU

Very closely related to Highway Networks

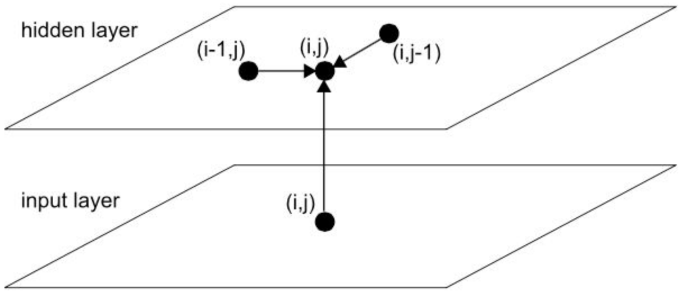


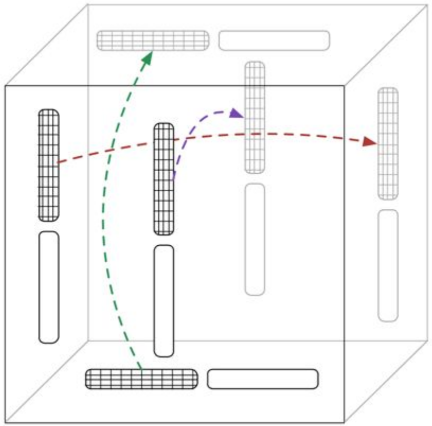
1d Grid LSTM Block

Grid-LSTM: 3D

3D Grid-LSTM = Multidimensional LSTM, but again with LSTM cells in depth dimension

2D Multidimensional RNN has 2 hidden vectors instead of 1

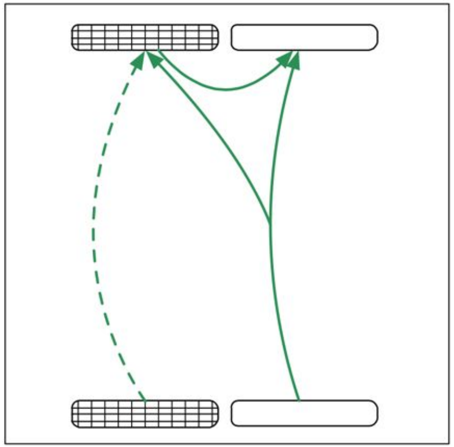




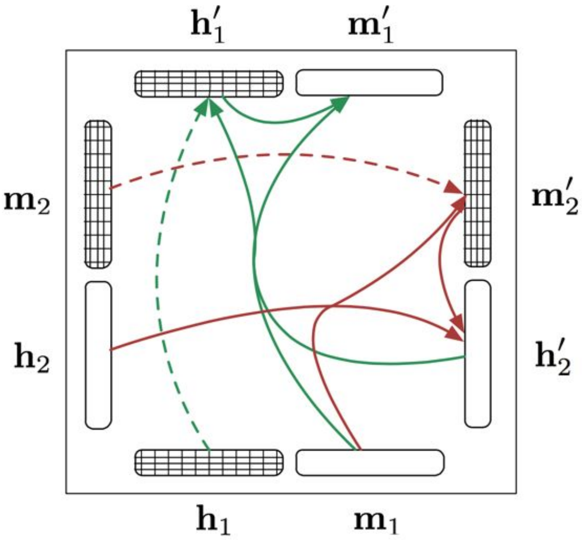
3d Grid LSTM Block

Grid-LSTM: All together now

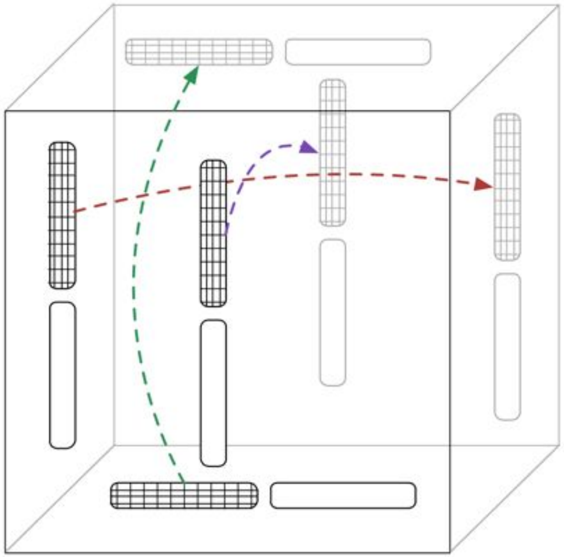
- N-D Grid-LSTM has N inputs and N outputs at each LSTM block



1d Grid LSTM Block



2d Grid LSTM block



3d Grid LSTM Block

Relation to Attention

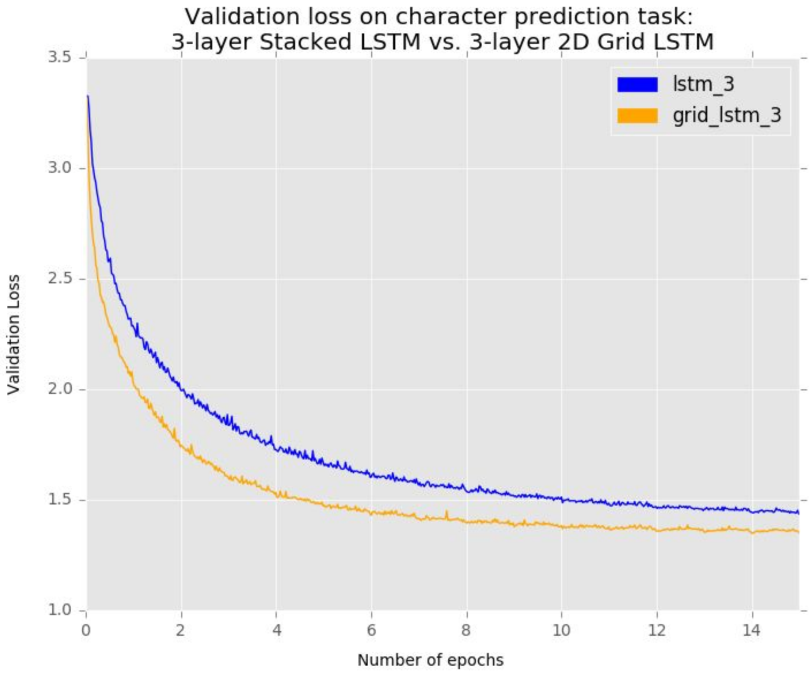
LSTM: “The mechanism also acts as a memory and implicit attention system, whereby the signal from some input xi can be written to the memory vector and attended to in parts across multiple steps by being retrieved one part at a time.” - Quoc Le

Grid-LSTM: “Another interpretation of the attention model is that it allows an O (T) computation per prediction step. So the model itself has O(T2) total computation (assuming the lengths of input and output sequences are roughly the same). With this interpretation, an alternative approach to the attention model is to lay out the input and output sequences in a grid structure to allow O (T2) computation. This idea is called Grid-LSTM” - Quoc Le

Experiment

Task: Character prediction

3-layer stacked LSTM vs. 3-layer stacked Grid-LSTM



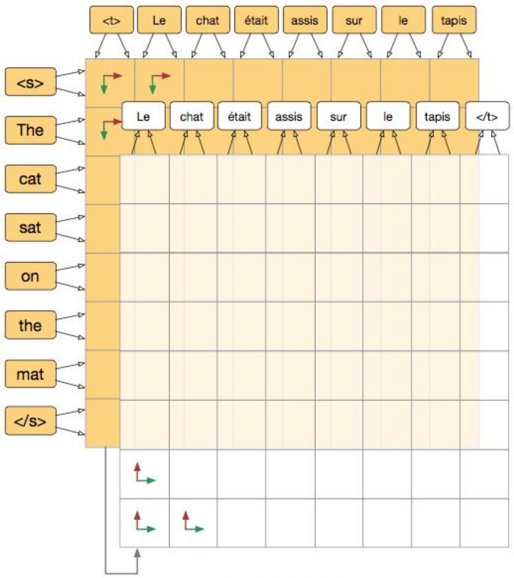
Future Work

Application to speech recognition, which uses stacked RNNs on spectrograms

- Start with 2D Grid-LSTM
- Can also try 3D Grid-LSTM

Machine translation

- 3D Grid-LSTM instead of encoder decoder network



3d Grid LSTM