Experiments

Dieuwke Hupkes

1 Datasets

I define the following set of languages:

Name	*Numeric leaves*	*Tree depth*	*Example*
L_2	2	1	$(x_1\ op\ x_1)$
L_3	3	2	$((x_1\ op\ x_2)\ op\ x_3)$
L_4	4	3	$((x_1\ op\ x_2)\ op\ (x_3\ op\ x_4))$
...			

Where $x_i \in \{-19, 19\}$, and $op \in \{+, -\}$. The meaning y of e sentences is the result of the arithmetic expression expressed by the languag. We restrict the languages to include only expressions such that $y \in \{-60, 60\}$.

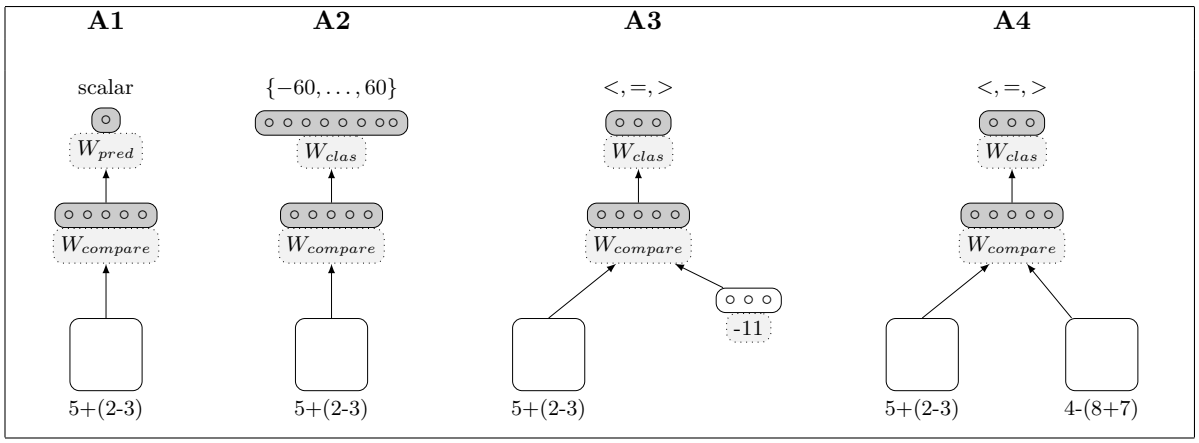
We define the following (structurally non-ambiguous) subsets of the languages defined above:

Name	*Restriction*	*Example*
L_i+	$op == +$	$(.(.(x_1 + x_2) + \dots x_i)$
L_i-	$op == -$	$(.(.(x_1 - x_2) - \dots x_i)$
$L_i$rb	only right branching trees	$(.(.(x_1\ op\ x_2)\ op\ x_3)\ op\ \dots x_i)$
$L_i$lb	only left branching trees	$(x_1\ op\ (x_2\ op\ (\dots\ op\ (x_{i-1}\ op\ x_i).).)$

The datasets that the networks will be trained and tested on are (subsets of) unions of the languages described above.

2 Architectures

I use four different architectures (explanation?):



3 Experiments

I will start by running a sequence of experiments to determine if the networks can learn to compose the meaning of sentences from the structurally non ambiguous languages L_2, L_3+, L_3-, L_3lb and L_3rb. Depending on the results I will move on to more complicated languages In principle, I would like to do all (possible) combinations that can be made by combining elements from the following table,[1] starting with architectures A1 and A2 and then expanding to A3 and A4.

Network	**Language**	**Architecture**	**Dimensionality**	**Initialisation**	**Embeddings**
SRN	L_2	A1	10	Random	fixed
GRU	L_3+	A2	6	Gray	trained
LSTM	L_3-	A3	2	one-hot?	
	L_3rb	A4			
	L_3lb				

4 More concrete plan

Wat moet ik doen?

- Generate datasets (or think of how to do that on the fly)
- Implement architectures
- Train networks and plot results

[1] Of course excluding non-sensical combinations, such as Gray encoding in two dimennions