Simulations and theory of generalization in recurrent networks

Juan Valle-Lisboa (juancvl@psico.edu.uy)
Centro de Investigación Básica en Psicología (CIBPSI), Facultad de Psicología & Sección Biofísica, Facultad de Ciencias UDELAR, URUGUAY

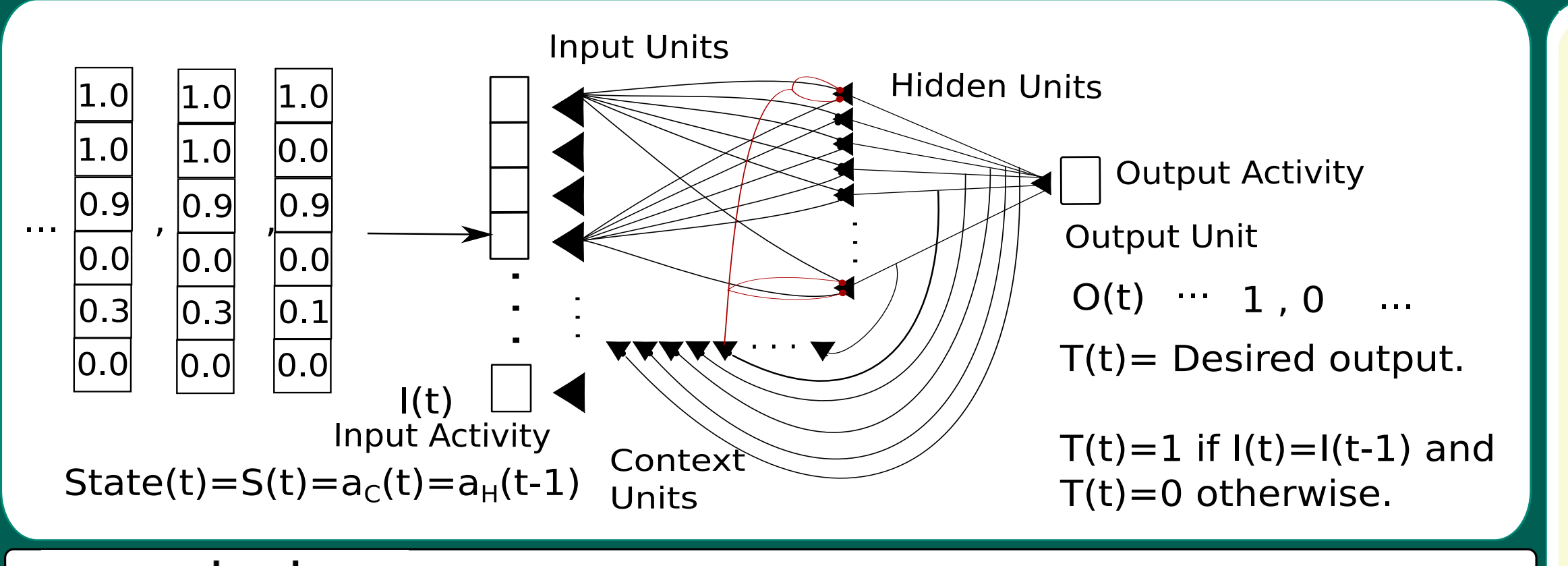
Introduction

Despite the tremendous advances of Artificial Intelligence, a general theory of intelligent systems, connecting the psychological, neuroscientific and computational levels is lacking. Artificial Neural Networks (ANN) are good starting points to build the theory. Although nowadays most ANN models are said to be essentially unrelated to the Brain Networks, they embody several principles that most Neuroscientists would agree are central to the Nervous System working.

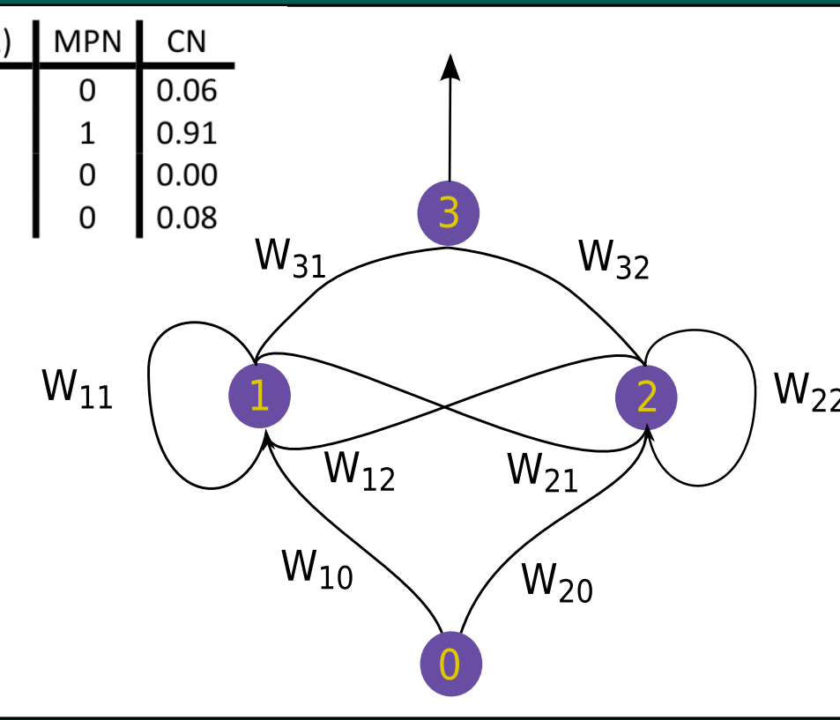
These features are massive parallelism, the storage of information superposed in a distributed set of synapses, that cells detect abstract features in a hierarchical fashion and that learning occurs by changing these synapses to reduce some measure of cost or error. Moreover, ANN are increasingly used to model Neuroscience data, with cells within the model that share several response properties of real neurons. The difficulties these models find when dealing with cognitive problems can point towards problems in the shared theoretical commitments of Neuroscience and ANN. In order to follow this path, we propose to analyze the generalization of learning in simple but challenging problems.

We have previously proposed to concentrate on learning sameness, as we have shown that this is difficult for a SRN. Here we present the results of trying to use a Long-Short Term Memory Network to learn sameness. We show that the LSTM although much more efficient to learn partial examples of sameness fails to generalize to a proportion of the examples. This suggests that LSTM and SRN share a core set of features that make generalization of sameness problematic. By analyzing where the two models fail, we arrive at a proposal of what makes sameness hard to learn and generalize in recurrent neural networks.

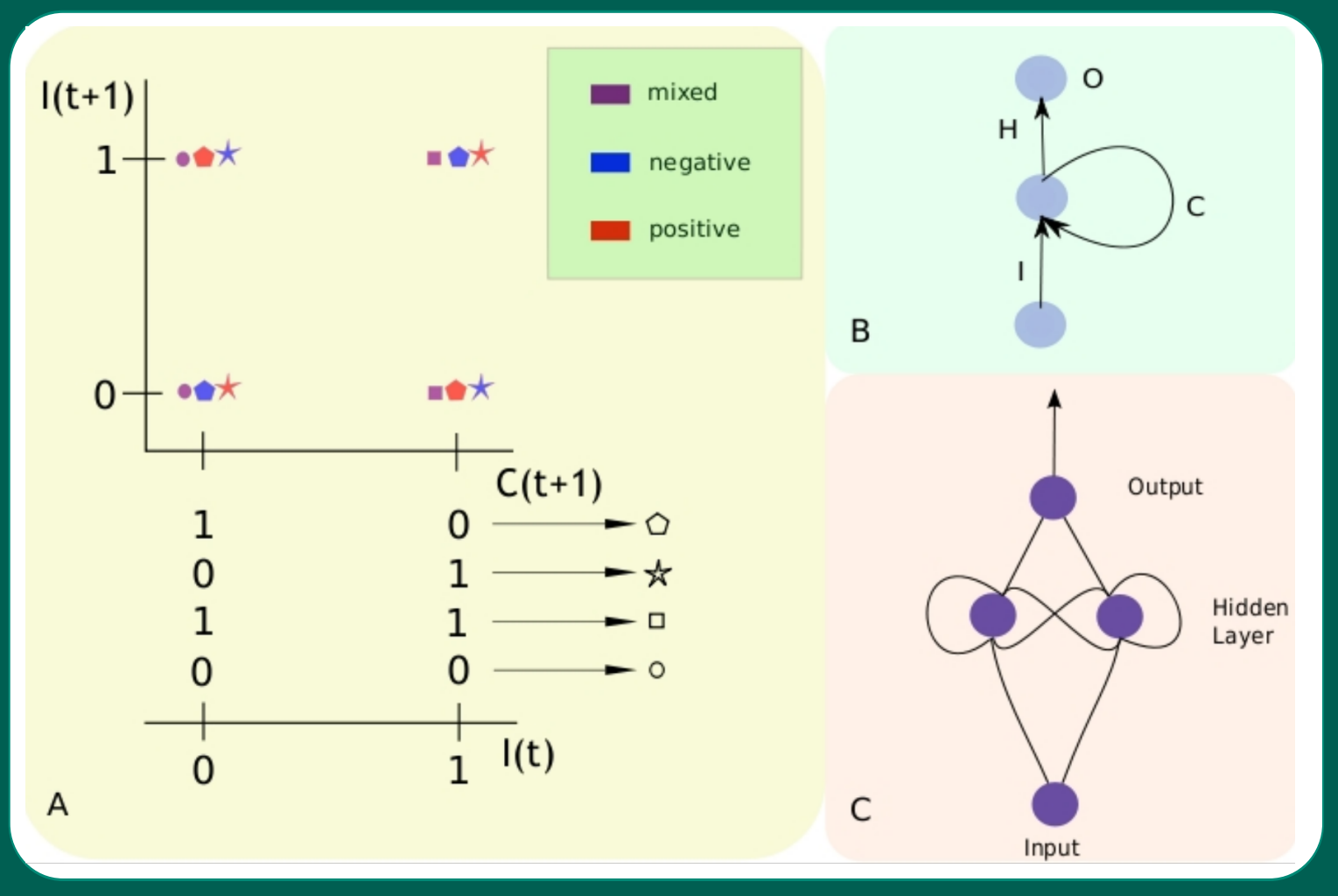
Sameness and what it takes to compute and learn it



I(1)	I(2)	MPN	CN
1	1	0	0.06
1	0	1	0.91
0	1	0	0.00
0	0	0	0.08



I(1)	I(2)	I(3)	MPN(1)	MPN(2)	CN(1)	CN(2)
1	1	1	1	0	0.98	0.02
1	1	0	1	1	1.00	0.97
1	0	1	1	0	0.98	0.02
1	0	0	1	1	1.00	0.97
0	1	1	0	0	0.03	0.00
0	1	0	1	0	0.98	0.02
0	0	1	0	0	0.02	0.00
0	0	0	1	0	0.97	0.02



The network in B) cannot compute temporal equivalence. A) Markers indicate the different combinations of contex (C(t+1)) in response to input I(t); the plane shows every combination of I(t+1) and context colored according to the desired output. Mixed=hidden unit does not distinguish the two possible inputs. The star and the pentagon mark distinguishable non linearly separable cases. C) Minimal SRN that can achieve the task.

General aspects of sameness

For general n-dimensional vectors, with k possible values, there are k^n vectors and k^{2n} pairs of vectors. For binary vectors k=2, a small size vector of dimension there are 2^{200} ~ 1.6 x 10^{60} pairs. Even presenting a pair per femtosecond would take much more than the age of the universe to experience them all. Another complication is that the fraction of pairs for which a positive answer is required decreases with size. In effect there are only k^n pairs out of k^{2n} , ie the fraction of pairs is k^{-n} . Thus bigger sizes imply less frequent pairs of vectors requiring a positive answer.
As can be seen, there are nevertheless possible implementations of sameness in neural networks.
The question is whether an ANN can learn it from (much less data).
I will use accuracy but witha a caveat: For 10 dimensions there are 1024 possible binary vectors and 1048576 pairs.A network outputing 0 for all examples would have an accuracy of 0.99902 and a mean absolute error of 9.8e-4.

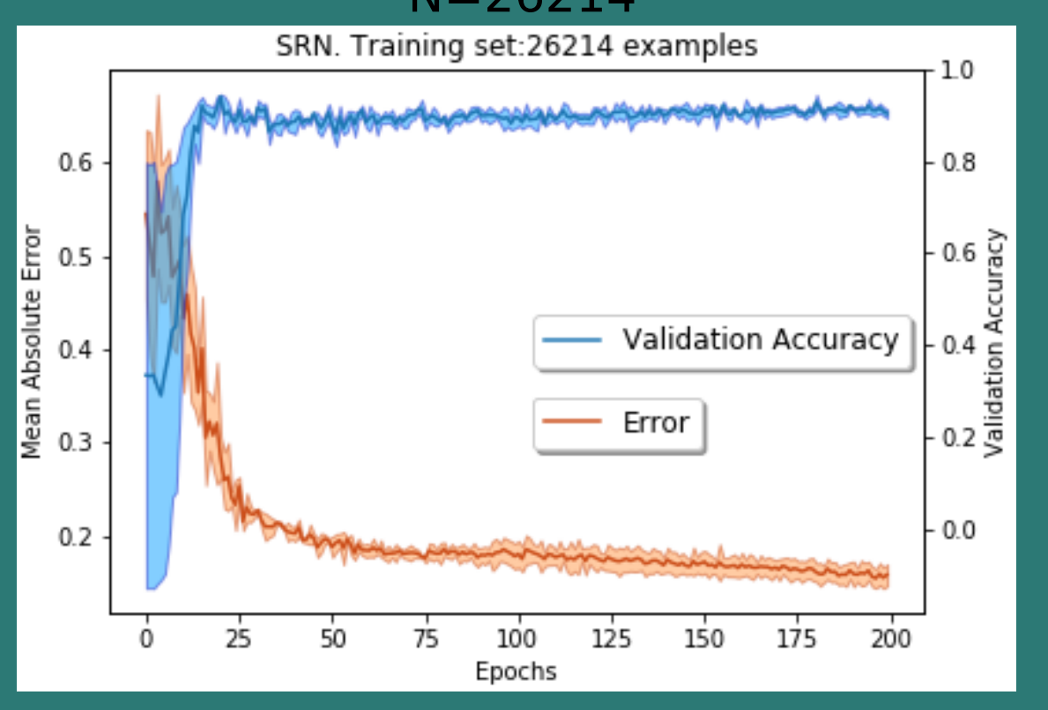
Methods

Using Tensorflow with the Keras backend, I trained several LSTM and SRN models to learn 10-dimensional sameness.
Each network had 30 hidden units, and 1 sigmoid output unit.
Networks were trained using the Adam optimizer, and we settled for a learning rate of 1e-3 (others led to instability or stalling).
Batch size was 128.
3 types of training sets: random sampling from the vectors, random sampling with 10 % probability of positives; random sampling wiht 50 % probability
Training set sizes of 26214 (2.5 % of all cases), 104857 (10 %), 1048576 (100 %).
Measured the training set error and accuracy, another set of pairs as validation set during training and evaluation in the whole set. We measuredd the accuracy for all positive cases ('all ones').

Simple Recurrent Networks

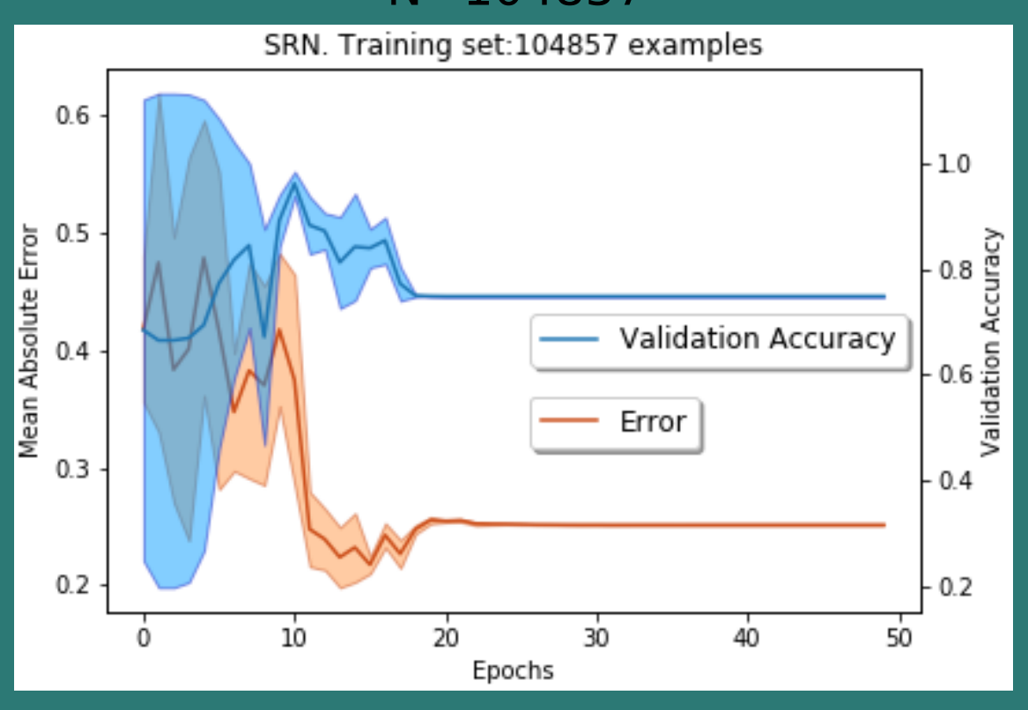
Random Sampling of pairs

N=26214



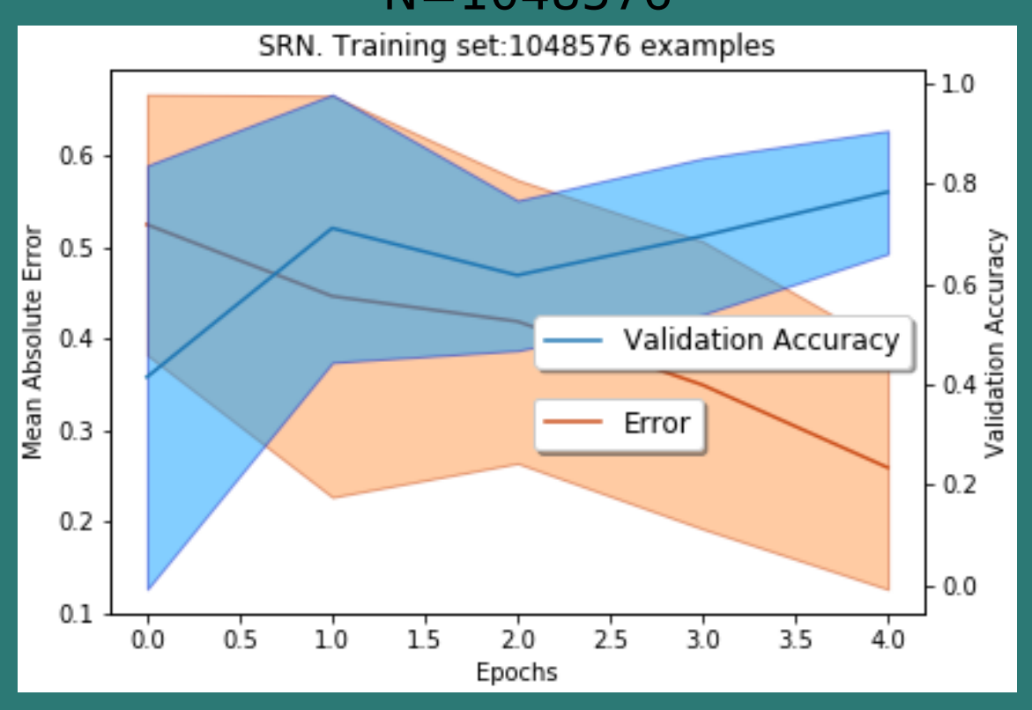
Number of false positives: 167772
Number of false negatives: 707

N=104857



Number of false positives: 262144
Number of false negatives: 512

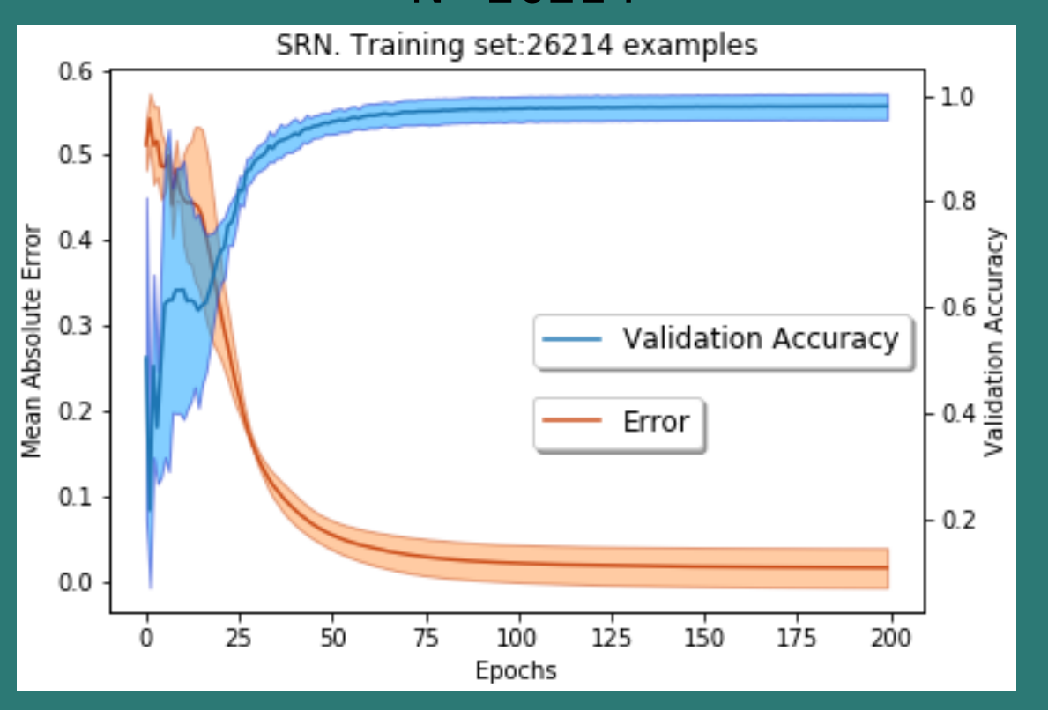
N=1048576



Number of false positives: 209715
Number of false negatives: 205

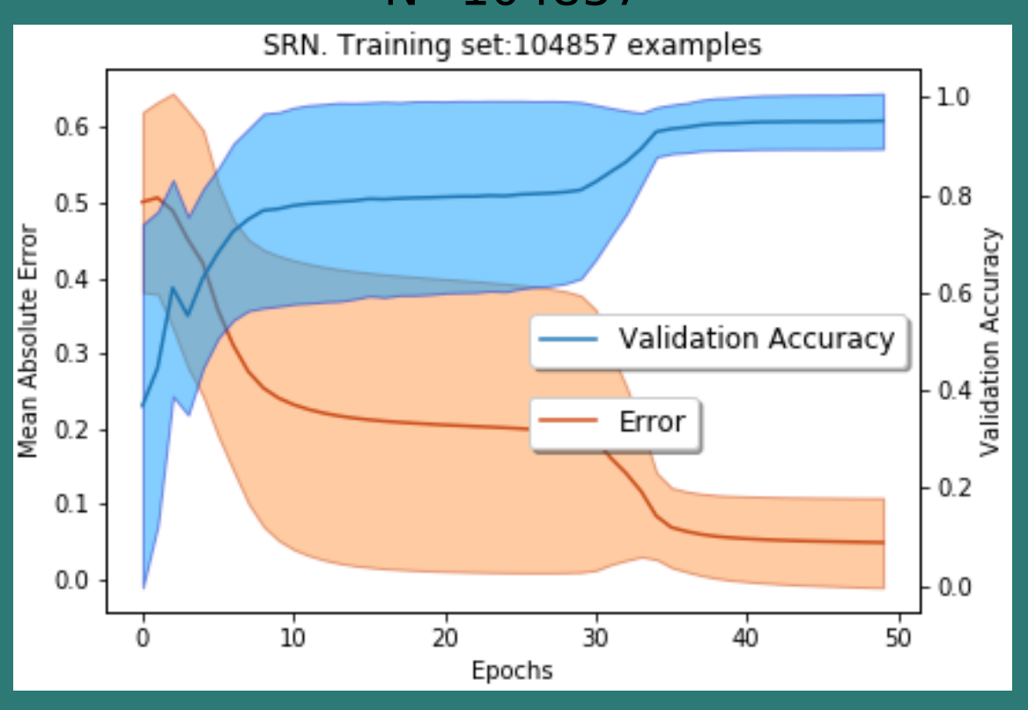
Ten percent of positive cases

N=26214



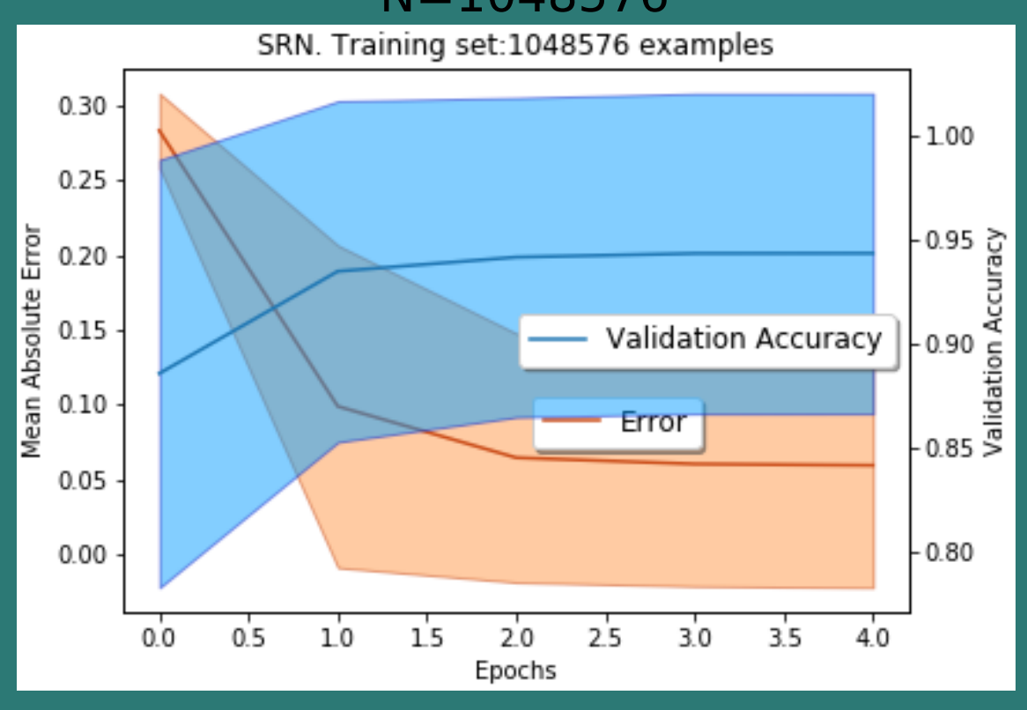
Number of false positives: 2412
Number of false negatives: 205

N=104857



Number of false positives:31457
Number of false negatives: 307

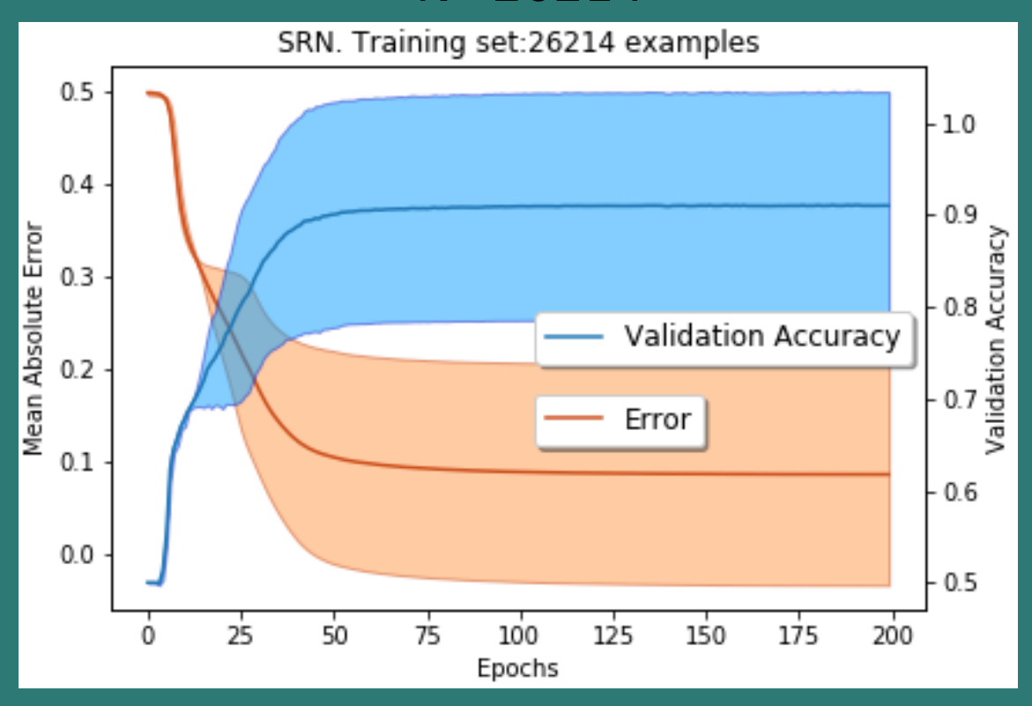
N=1048576



Number of false positives: 52429
Number of false negatives: 205

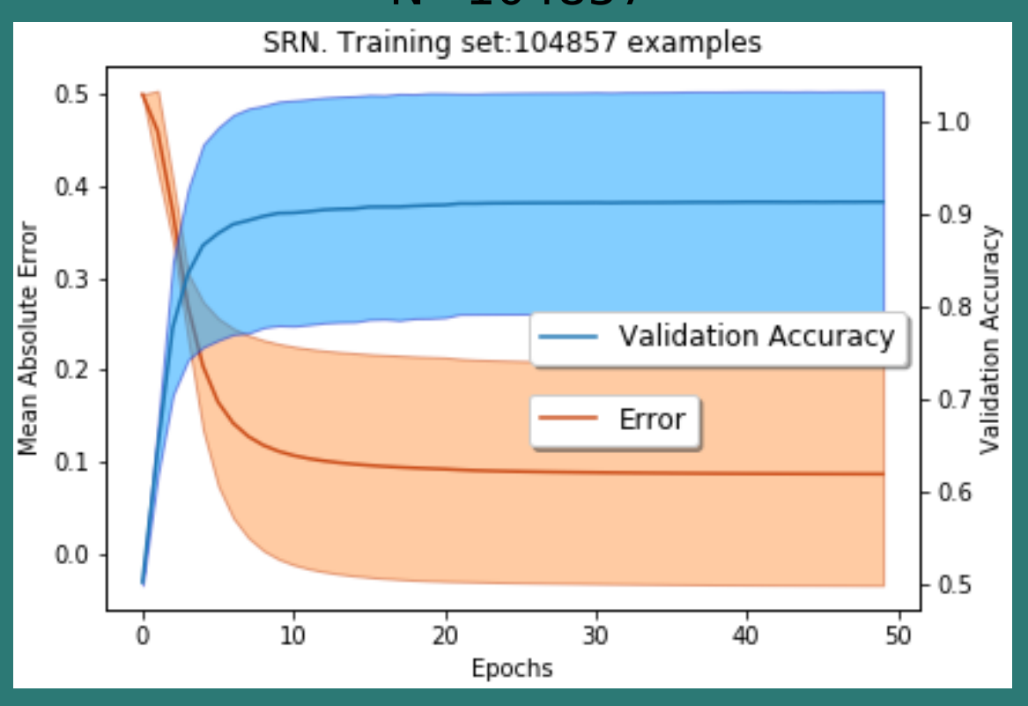
Fifty percent of positive cases

N=26214



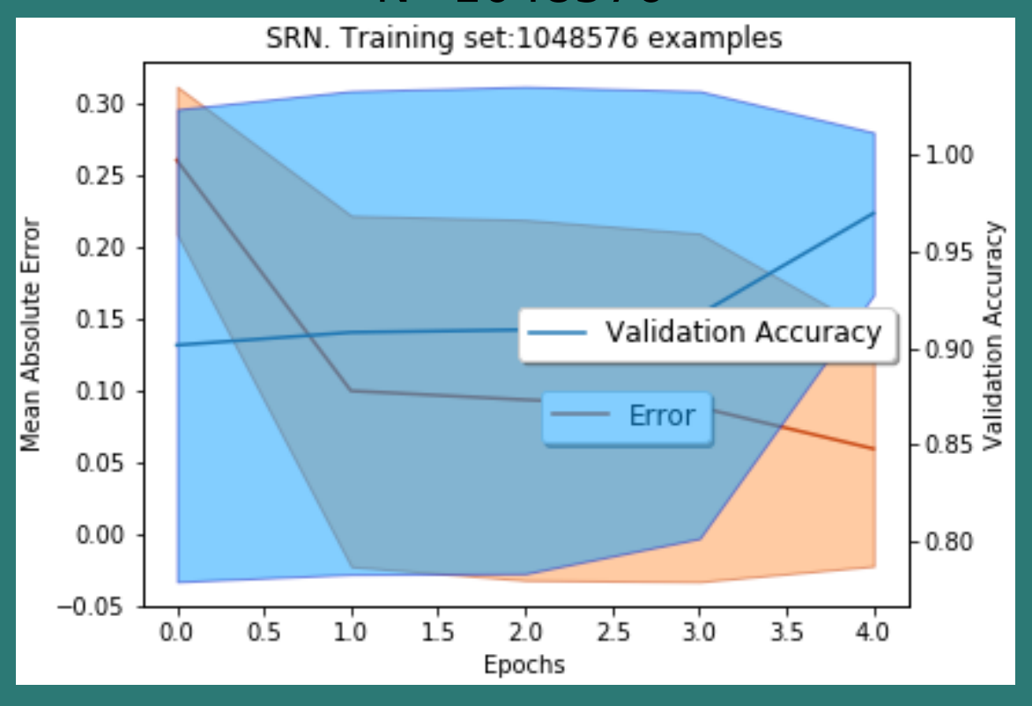
Number of false positives: 209715
Number of false negatives: 0

N=104857



Number of false positives: 209715
Number of false negatives: 0

N=1048576

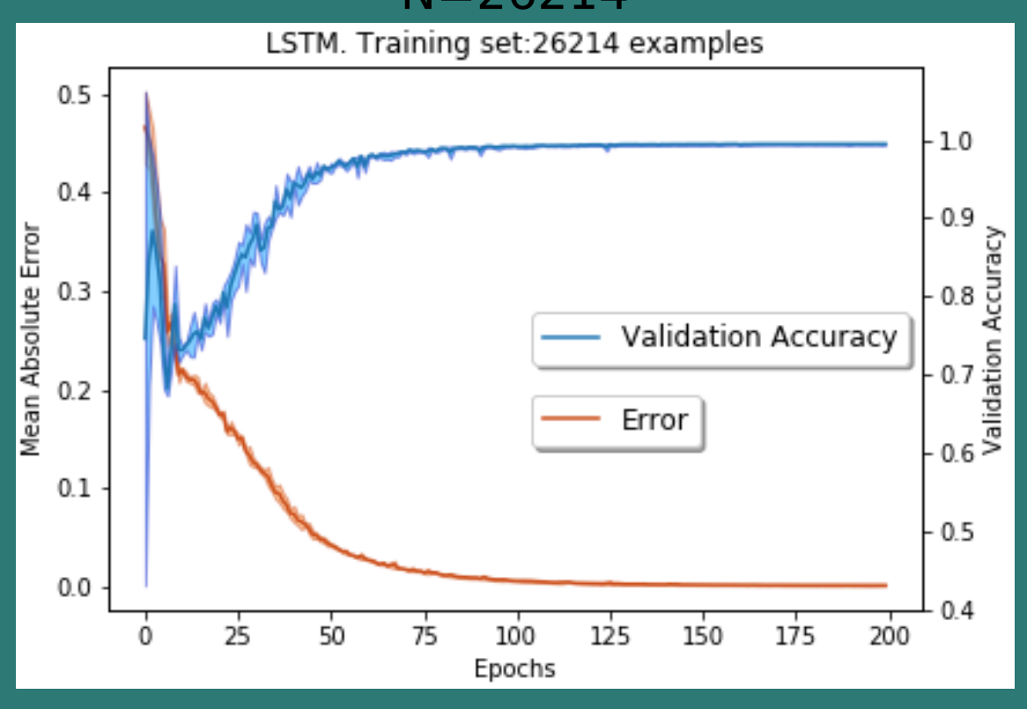


Number of false positives: 62915
Number of false negatives:0

Long Short Term Memory Networks

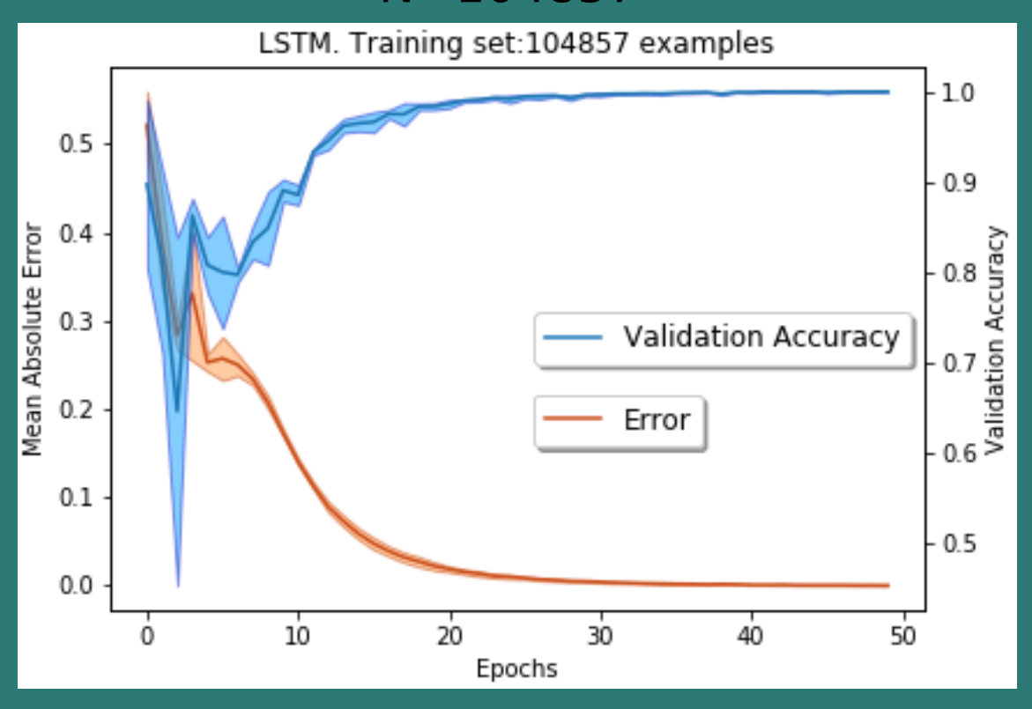
Random Sampling of pairs

N=26214



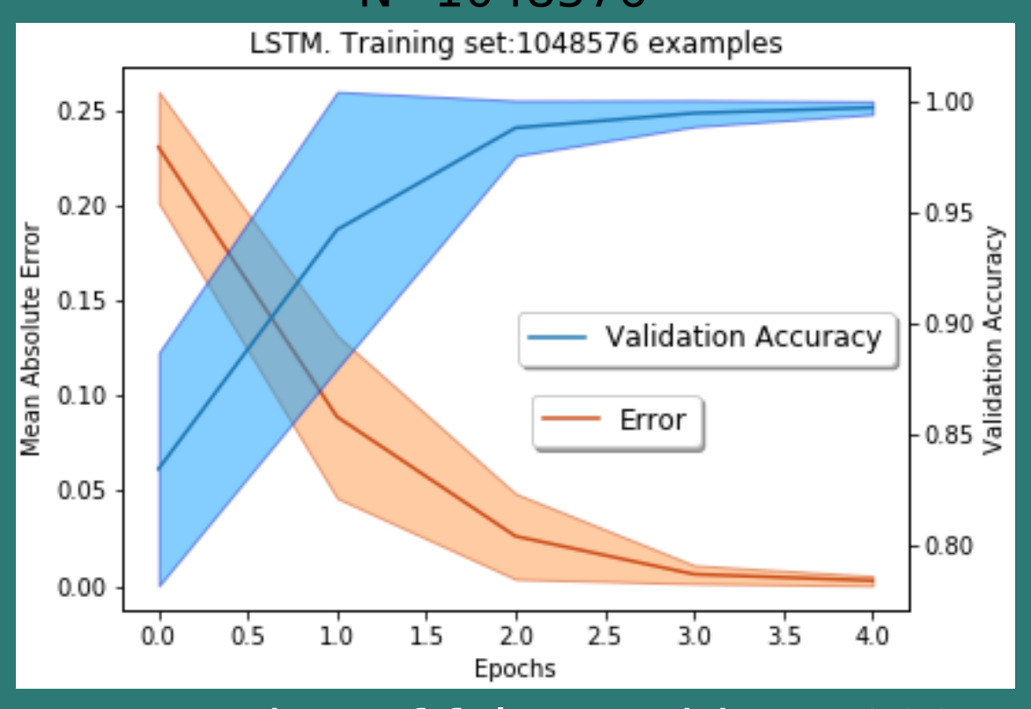
Number of false positives: 1783
Number of false negatives: 839

N=104857



Number of false positives: 325
Number of false negatives: 7

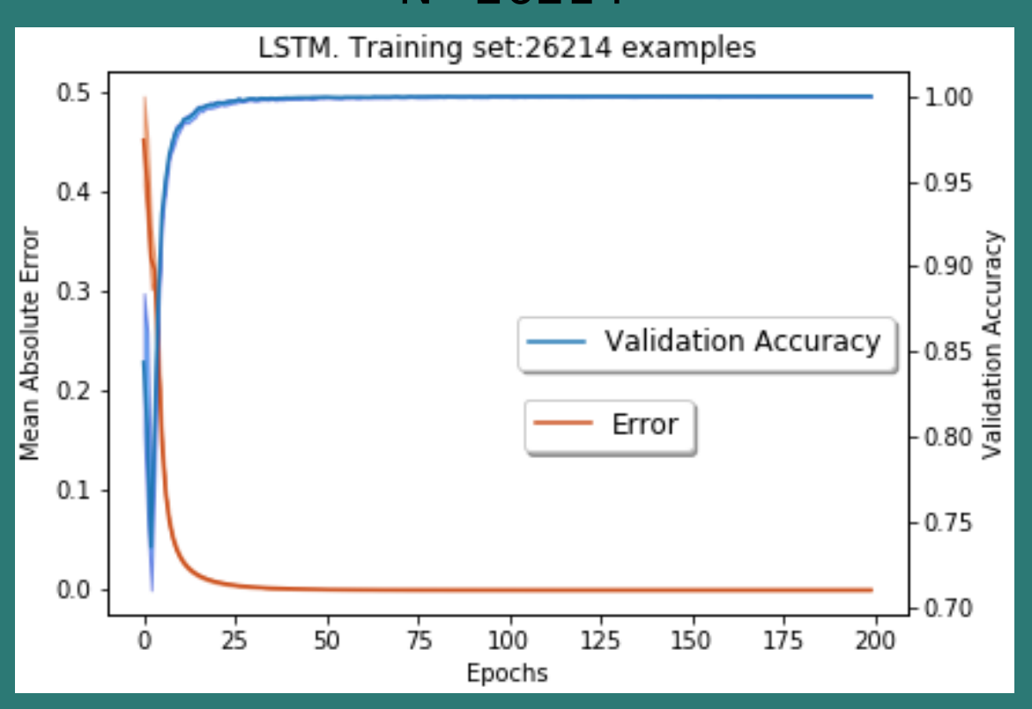
N=1048576



Number of false positives: 2097
Number of false negatives: 205

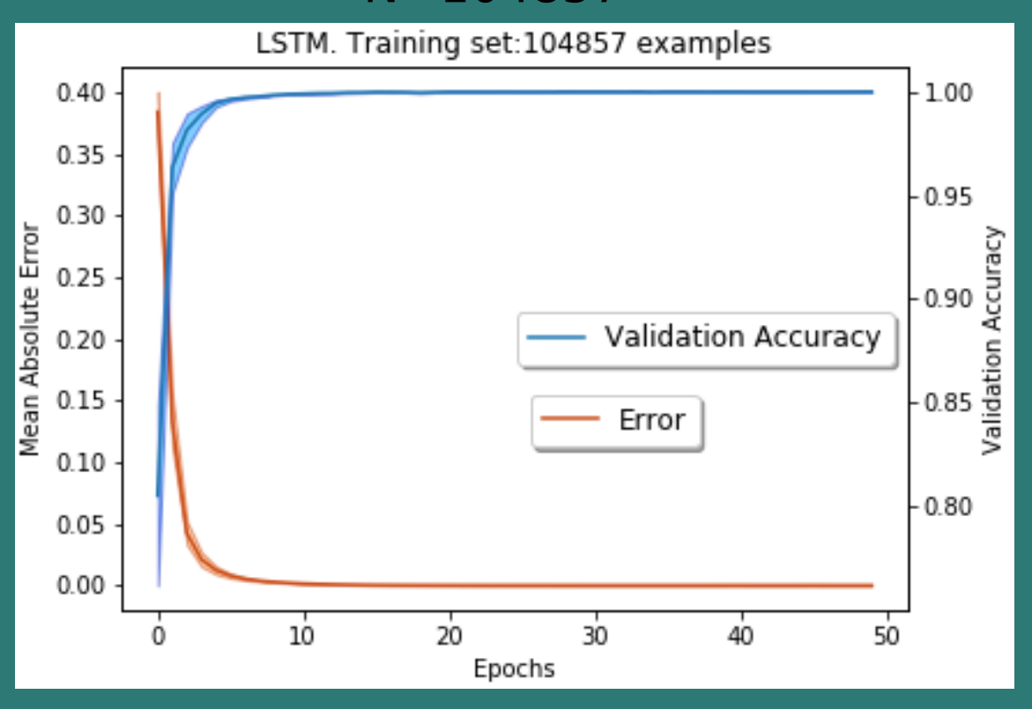
Ten percent of postive cases

N=26214



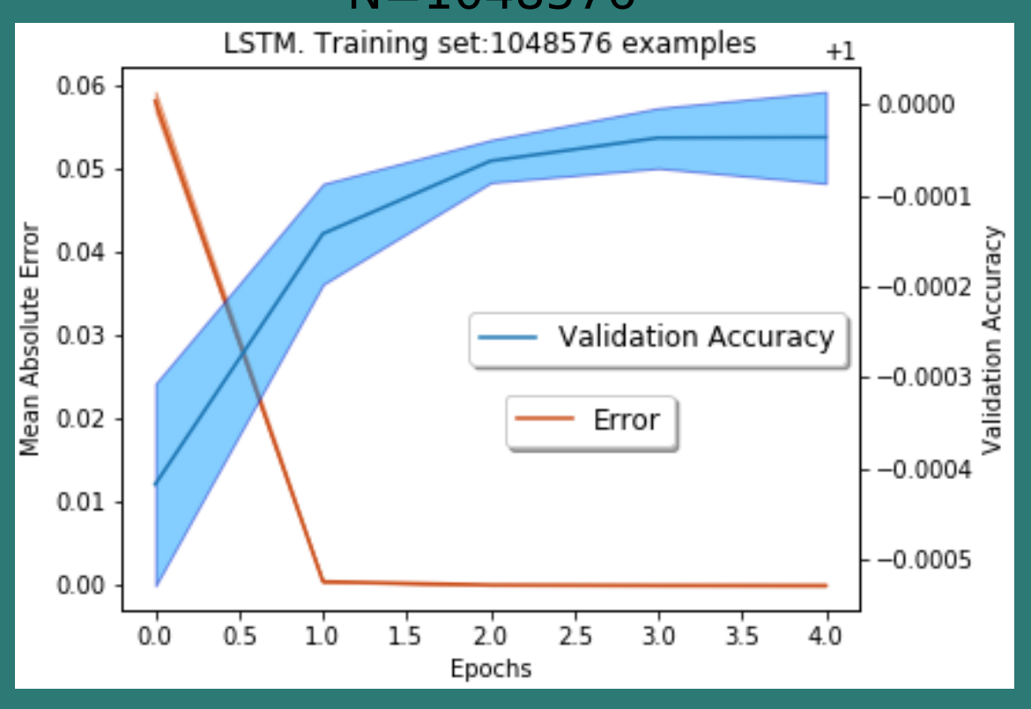
Number of false positives: 553
Number of false negatives:0

N=104857



Number of false positives: 94
Number of false negatives:0

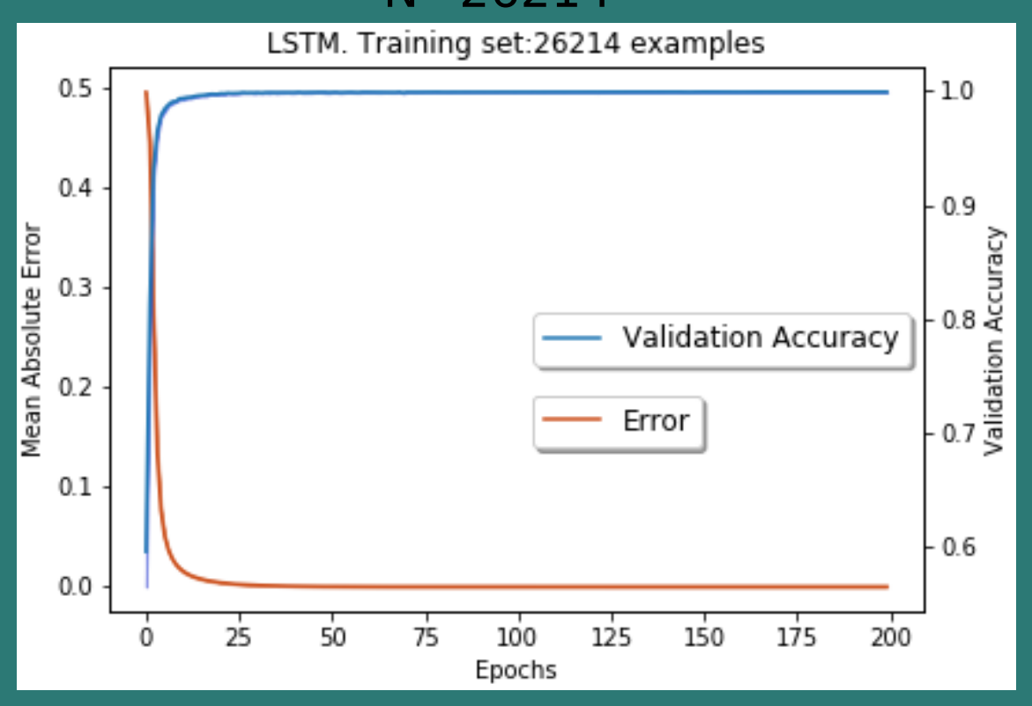
N=1048576



Number of false positives: 25
Number of false negatives: 0

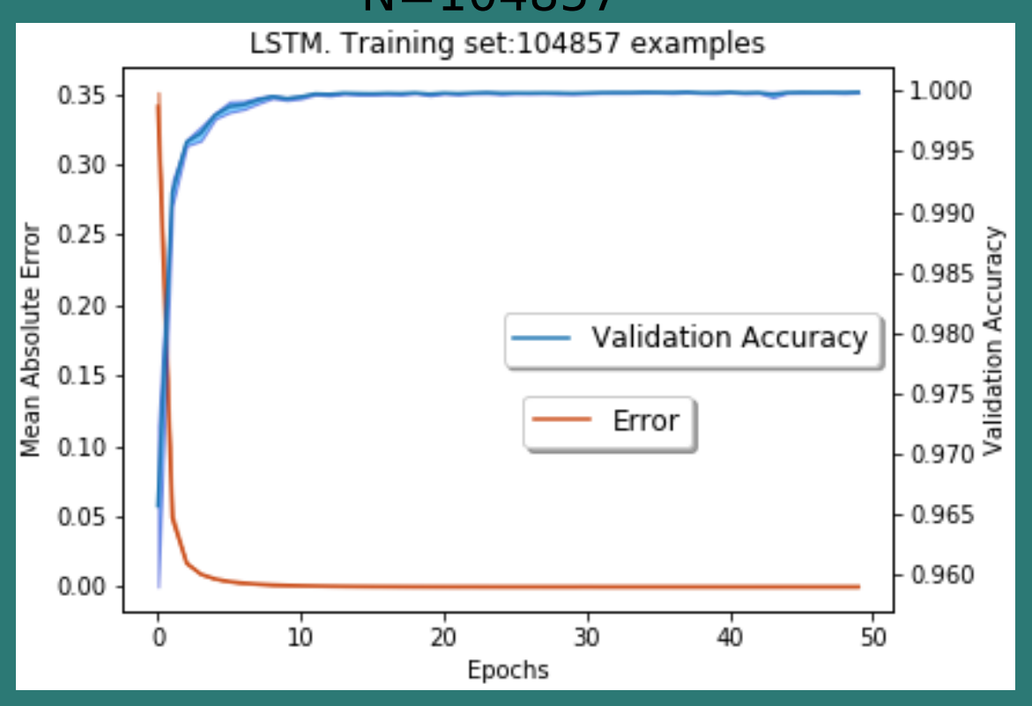
Fifty percent of positive cases

N=26214



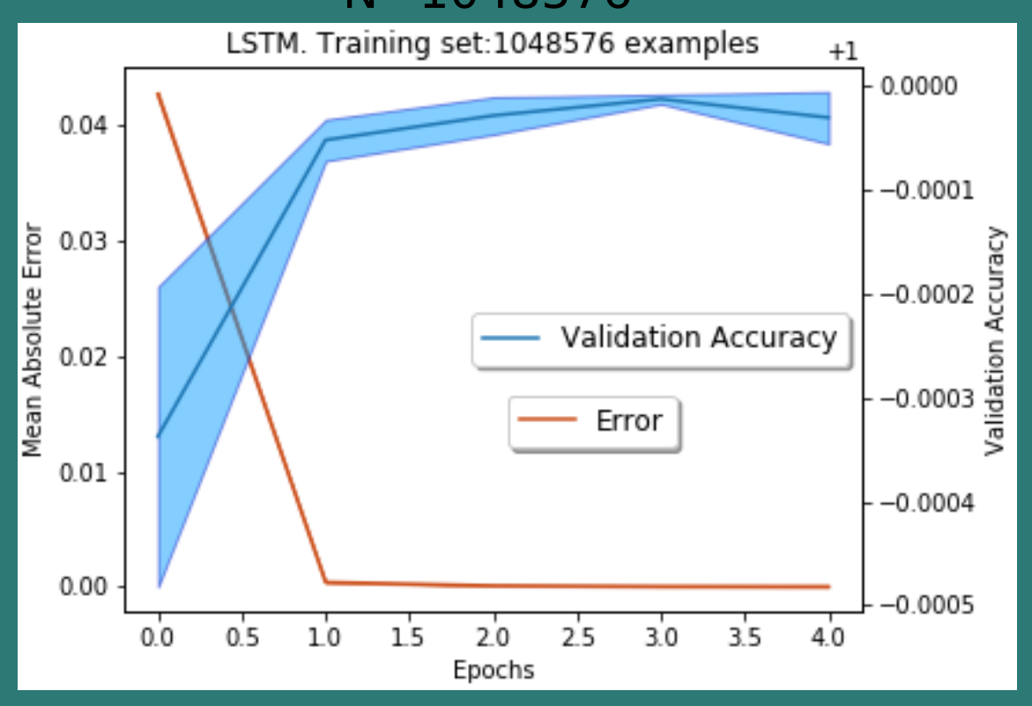
Number of false positives: 1363
Number of false negatives:0

N=104857



Number of false positives: 262
Number of false negatives: 0

N=1048576



Number of false positives: 63
Number of false negatives: 0

Summary of the Training and Testing Results

Type	Epochs	Training pairs	Fraction of positives	Training accuracy	Test Accuracy	All ones accuracy	Complete set accuracy
LSTM	5	1048576	0.1	0.999988 (1e-6)	0.99996 (5e-5)	1.000000 (<1e-6)	0.999976 (3e-6)
LSTM	5	1048576	0.5	0.999993 (<1e-6)	0.99997 (2e-5)	1.000000 (<1e-6)	0.99994 (2e-5)
LSTM	50	104857	0.1	1.000000 (<1e-6)	0.99988 (7e-5)	1.000000 (<1e-6)	0.99991 (2e-5)
LSTM	50	104857	0.5	1.000000 (<1e-6)	0.99989 (2e-5)	1.000000 (<1e-6)	0.99975 (4e-5)
LSTM	50	104857	1/1024	0.99987(5e-5)	0.99957 (6e-5)	0.993 (0.004)	0.99969 (5e-5)
LSTM	200	26214	0.1	1.000000 (<1e-6)	0.99969 (9e-5)	1.000000 (<1e-6)	0.999473 (9e-5)
LSTM	200	26214	0.5	1.000000 (<1e-6)	0.9992 (0.0003)	1.000000 (<1e-6)	0.9987 (0.0002)
LSTM	200	26214	1/1024	0.99903 (4e-5)	0.99481 (0.0002)	0.181 (0.008)	0.9983 (0.0001)
LSTM	5	1048576	1/1024	0.997 (0.003)	0.997 (0.002)	0.8 (0.2)	0.998 (0.001)
SRN	200	26214	0.1	0.98 (0.02)	0.98 (0.03)	0.8 (0.3)	0.9977 (0.0004)
SRN	50	104857	0.1	0.95 (0.02)	0.95 (0.02)	0.7(0.1)	0.97 (0.01)
SRN	5	1048576	0.1	0.94(0.09)	0.94(0.08)	0.8(0.2)	0.95(0.07)
SRN	5	1048576	0.5	0.94 (0.09)	0.97 (0.04)	1.000000 (<1e-6)	0.94 (0.09)
SRN	200	26214	1/1024	0.84 (0.01)	0.904 (0.007)	0.31 (0.01)	0.84 (0.01)
SRN	50	104857	0.5	0.9(0.1)	0.9 (0.1)	1.000000 (<1e-6)	0.8 (0.3)
SRN	200	26214	0.5	0.9(0.1)	0.9 (0.1)	0.9997 (0.0002)	0.8 (0.3)
SRN	5	1048576	1/1024	0.7 (0.1)	0.8 (0.1	0.8 (0.3)	0.8 (0.1)
SRN	50	104857	1/1024	0.748858 (<1e-6)	0.748055 (<1e-6)	0.500000 (<1e-6)	0.750000 (<1e-6)

Discussion and Conclusions

Sameness is hard to learn for the usual recurrent ANN models, specially when the training set is randomly sampled from the set of possible input vectors.
Sameness-computing networks exist that can be desgined by hand and compute sameness with a high degree of precission.
Yet, for a toy problem of 10-dim binary vectors, the equivalent of at least 5 times the number of total pairs (>1e6) is needed to train a LSTM to a high level of accuracy. Increasing the proportion of positive cases increases the likelihood of finding a network that ihas a high accuracy. This high accuracy is not perfect though. There are some false positives and false negatives. LSTMs are far superior than SRNs. This cannot be attributed to gradient explosion or extinction problems (given they are two-step dependencies).
Given that LSTM units are more complex, including the possibility to learn to store a value, this might be important to compute equivalence through time. In a sense, LSTMs have memory capacities similar to what authors like Gallistel & King (2009) postulate for neurons.
We need to analyze these cases in more detail to understand the differences from a detailed theoretical perspectives
In particular we want to understand whether these more powerful units embody a form of innate knowledge that can be conceived as a meta-rule innate capacity..
The computation of sameness is pervasive in humans and can be taught to animals, even bees. Given the difficulties in training even small sets, there are few open possibilities; that animals (including humans) just compute an approximate sameness much easier to learn, but prone to error; that there are other unknown learning mechanisms that are more powerful and require less experience; that sameness is part of our innate endowment.

Acknowledgments

I received partial support from CSIC -UDELAR and ANII.
Ithank Alvaro Caban for helpful comments.