IT'S ALL ABOUT THE NETWORKS

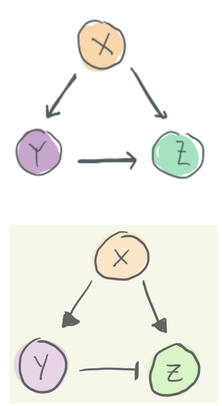
INNA SHAPOVALENKO (STOLNE SCHOOL), ARINA SHIGAEVA (LAUDE SAN PEDRO INTERNATIONAL COLLEGE), LIM YU CHENG (DAMANSARA JAYA MIDDLE SCHOOL), VERA TERENTEVA (ITMO UNIVERSITY), LAURA AVINYO (AMALFI ANALYTICS)

Our goal is to generate network motifs temporal data.
There are two ways to approach it 1. Mathematical simulation 2. Machine learning.
Here we present a mix of both. A NN using LSTM and a Basic NN with integration step

FEEDFORWARD LOOP (FFL)

COHERENT FFL TYPE 1

Coherent FFL is where both transcription factors X and Y are transcriptional activators. If the direct path to Z is the same as the indirect path to Z - we have a coherent loop.



INCOHERENT FFL TYPE 1

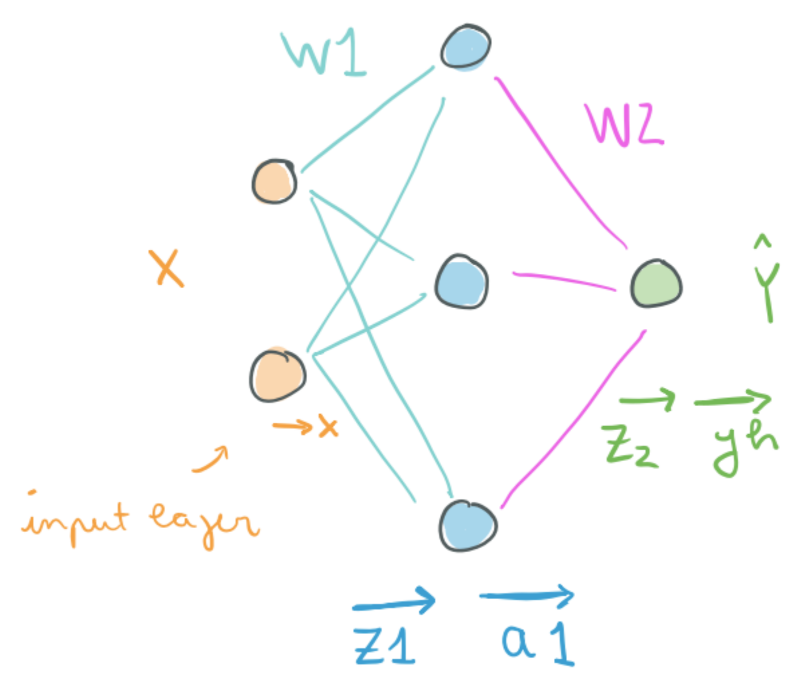
Incoherent FFL is where the direct path to Z acts in opposition to the indirect path to Z. In this case, X activates Y and Z, while Y represses Z.

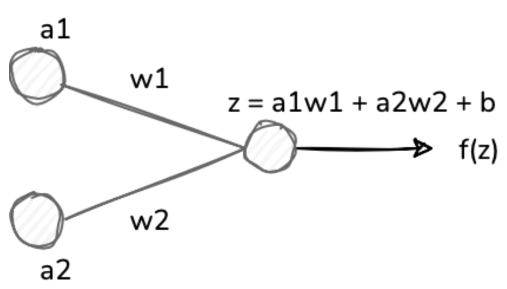
NETWORK NETWORKS (NN)

FORWARD FUNCTION

We used the forward function to transform the input into the output. To do so we follow the next steps:

1. The input Layer gets x and gives it to the net.
2. As each line has its weight w, we need to multiply the net x by the weight w of the line (we will have x*w).
3. We need to sum up all values of the incoming links (e. g. Z = x1*w1 + x2*w2)
4. Then we apply the activation function f(x), to Z, so we get f(Z).
5. We do the same steps for all hidden layers
6. The last activation function will return the final result





a1 = 1 a2 = 1	f(z) = 1	w1 = w3 = w
a1 = 1 a2 = 0	f(z) = 0	w >> 0
a1 = 0 a2 = 1	f(z) = 0	
a1 = 0 a2 = 0	f(z) = 0	b = -1.5w

BACKPROPAGATION

We use backpropagation to find the set of weights that has less error.

1. We get the predictions and compute how big is the error.
2. We tweak the weights slightly in the direction we will decrease the error.
3. We repeat till we get the best set of weights (ones that have less error).

NEURAL NETWORK AS GENETIC SYSTEM SIMULATOR

STEP 0: Prepare the network.
Set the parameters of the system.
Set the Weights and Biases of the network.
Following the same method as in the figure below.

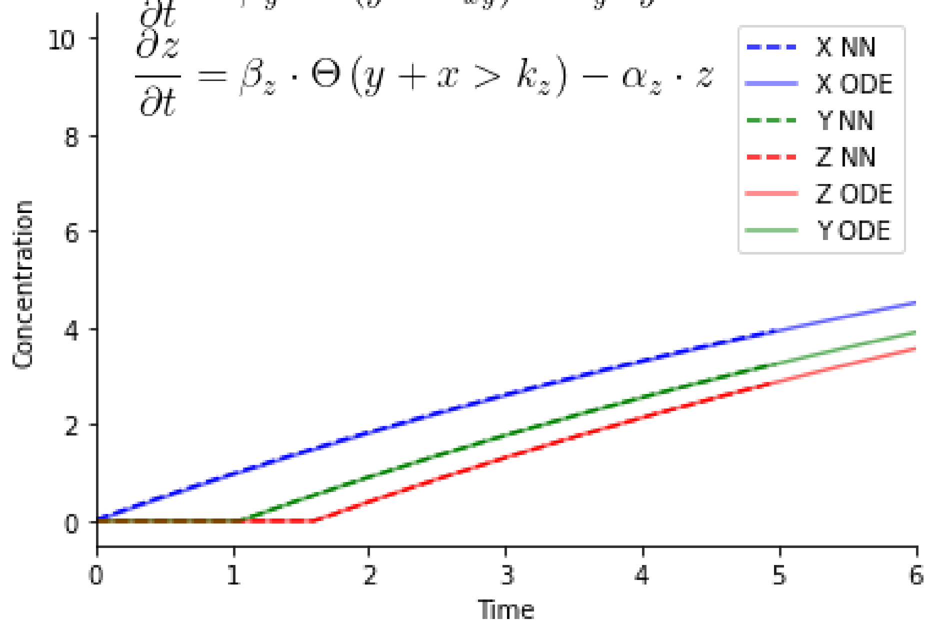
STEP 1: X, Y, Z are going to be produced?
Using the Forward function. That behaves as the Theta

STEP 2: How much X, Y, Z will get?

$$\hat{x} = x + h \cdot \frac{\partial x}{\partial t} \quad \hat{y} = y + h \cdot \frac{\partial y}{\partial t} \quad \hat{z} = z + h \cdot \frac{\partial z}{\partial t}$$

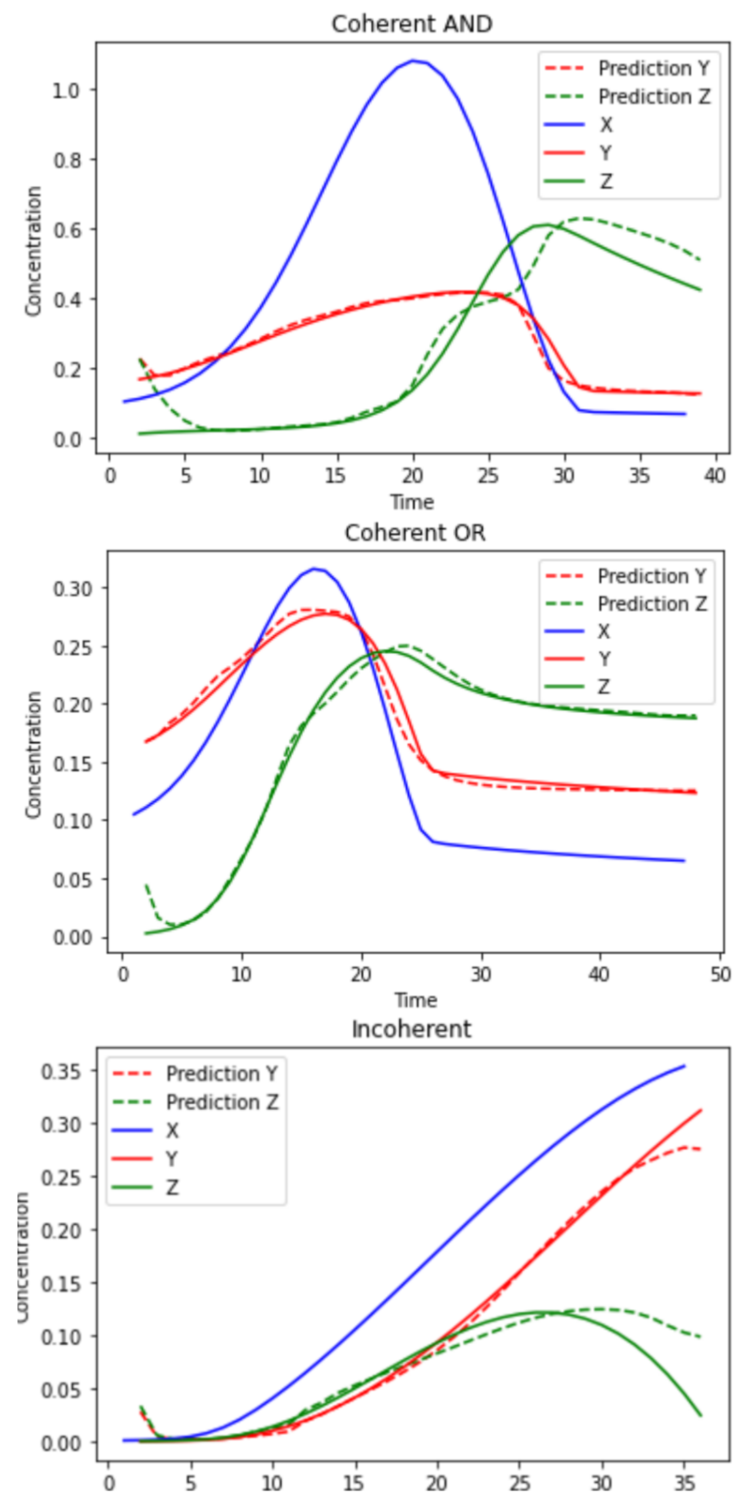
Repeat STEPS 1 and 2 as many times as necessary

$$\frac{\partial x}{\partial t} = \beta_x \cdot \Theta\left(x > k_{ux}\right) - \alpha_x \cdot x$$
$$\frac{\partial y}{\partial t} = \beta_y \cdot \Theta\left(y > k_{xy}\right) - \alpha_y \cdot y$$
$$\frac{\partial z}{\partial t} = \beta_z \cdot \Theta\left(y + x > k_z\right) - \alpha_z \cdot z$$

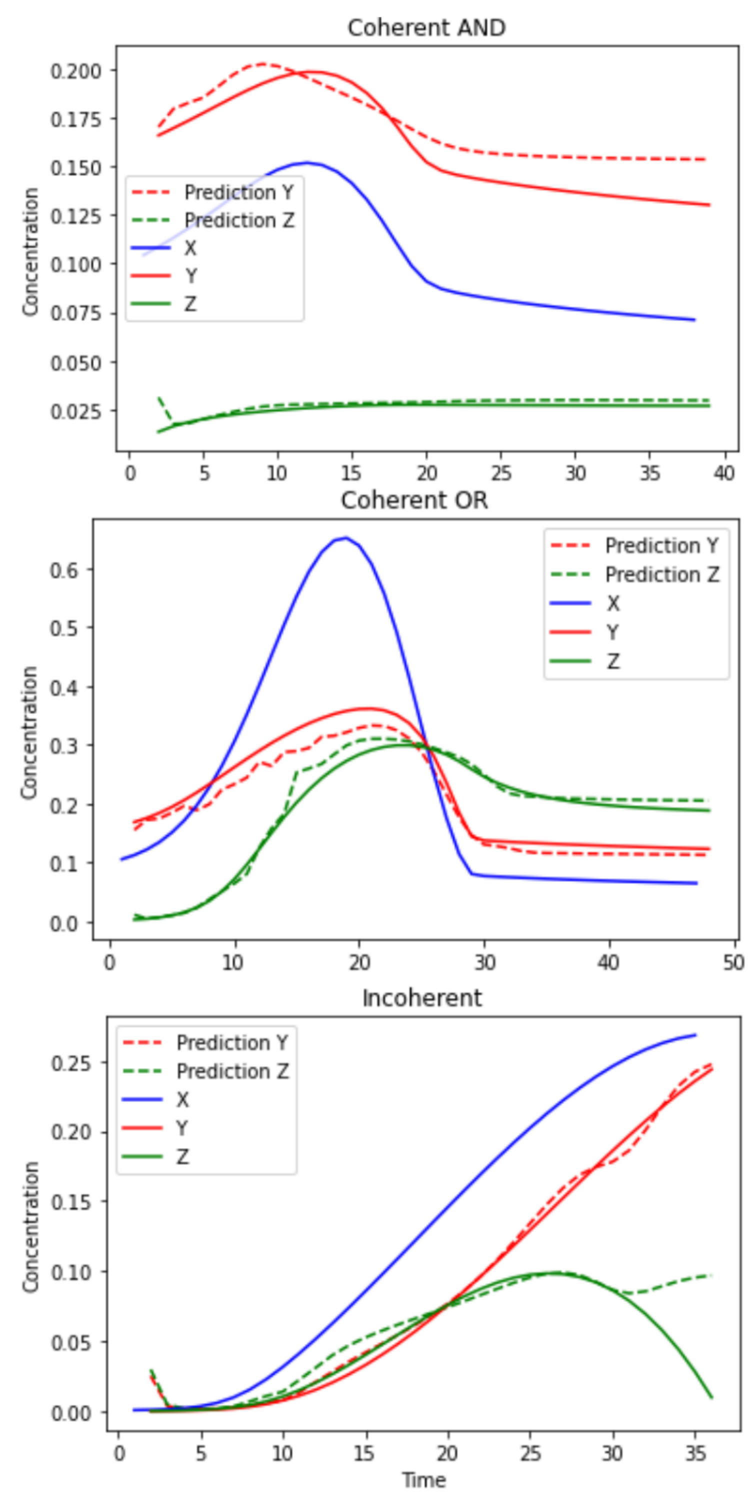


LONG SHORT TERM MEMORY

TRAINING



TEST



CONCLUSIONS

1. LSTMs are prospective candidates for simulating genetic systems. However, a better training algorithm and architecture is needed!
2. Basic Neural Networks well design can behave as Genetic systems. Particularly, defining where or not the a gene is going to be produce. Allowing us to easily parallize an other wise computer demanding operation.

IMPROVEMENTS TO LSTM

1. Automation by adjusting dynamically the learning rates and the number of memory blocks and cells
2. Pick the weights
3. Implement early stopping