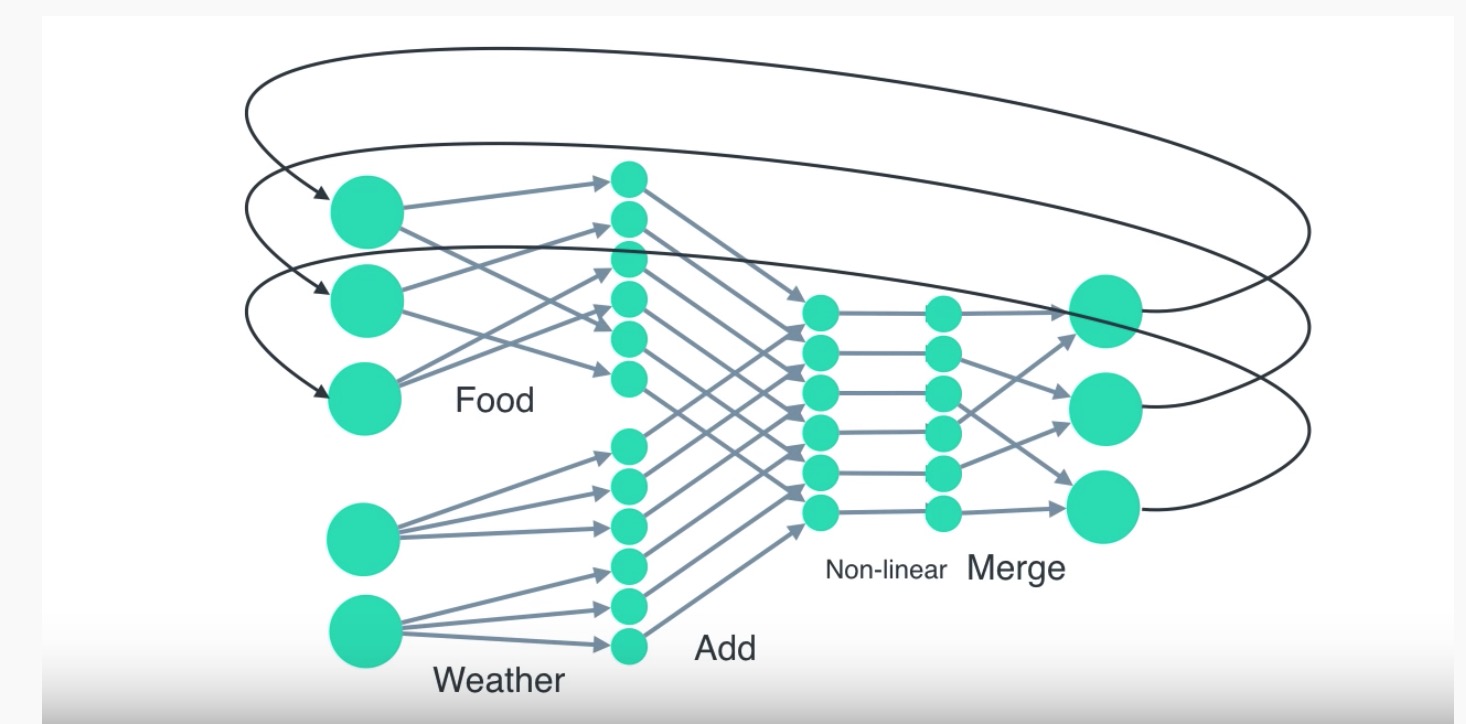
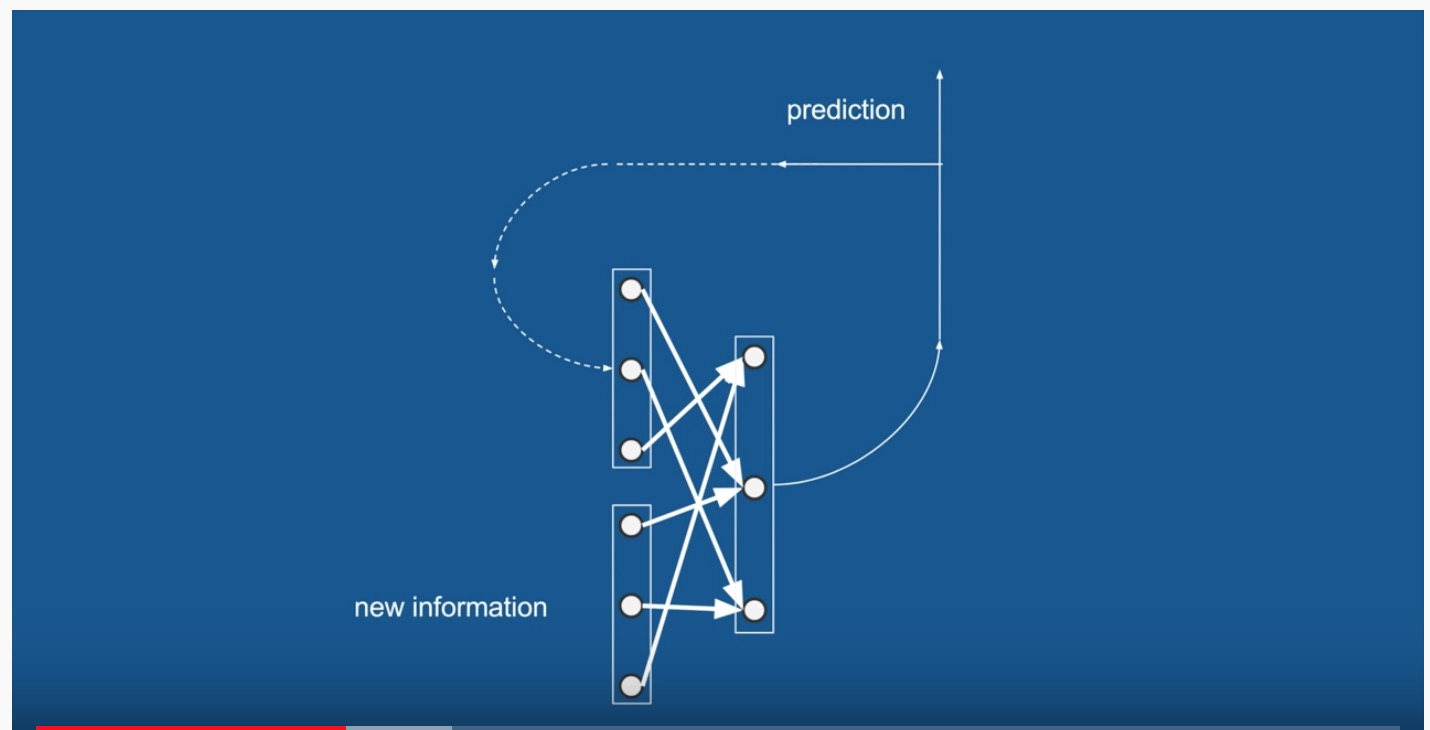
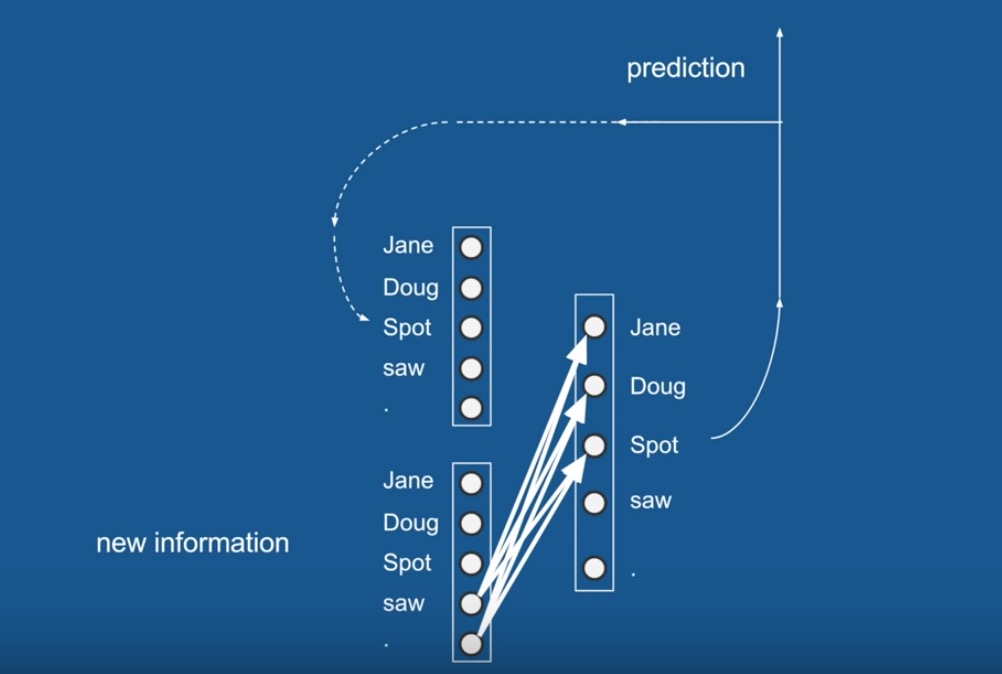
Learnings:

**Intro to RNN**: <https://www.youtube.com/watch?v=UNmqTiOnRfg>

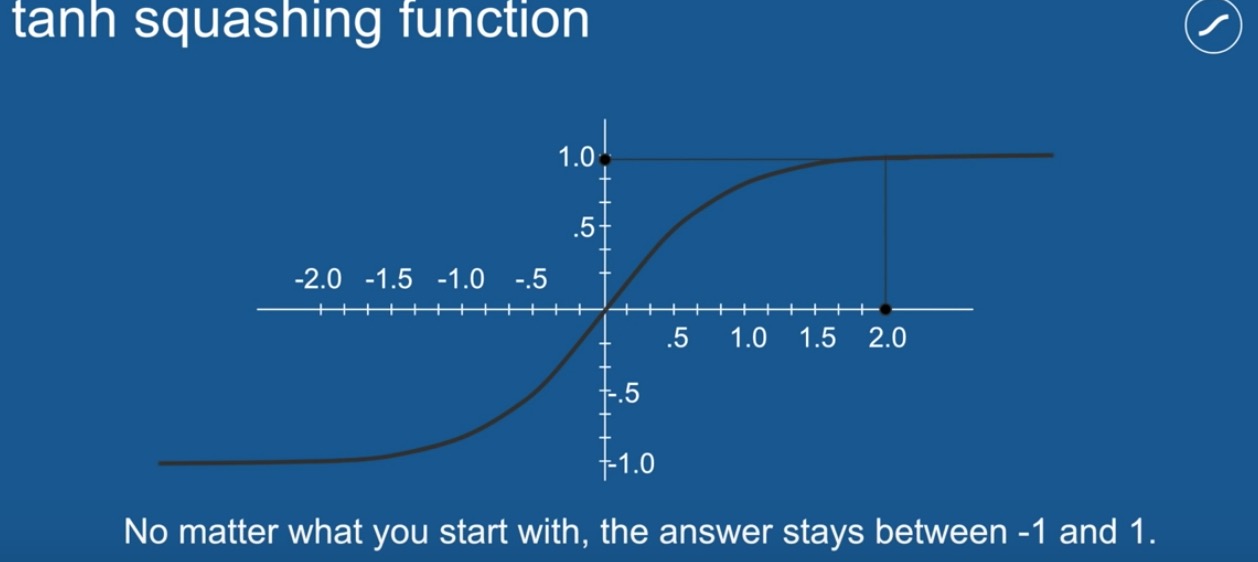
* Recurrent NN – output goes back in as input – input is the same as output that came from previous step
* RNN very useful when data is sequential.
* 
* how to train RNN – start with random weights – train using gradient decent –
* feed in data into NN and produces error – do steps to reduce error slowly –

<https://www.youtube.com/watch?v=WCUNPb-5EYI> - **RNN + LSTM**

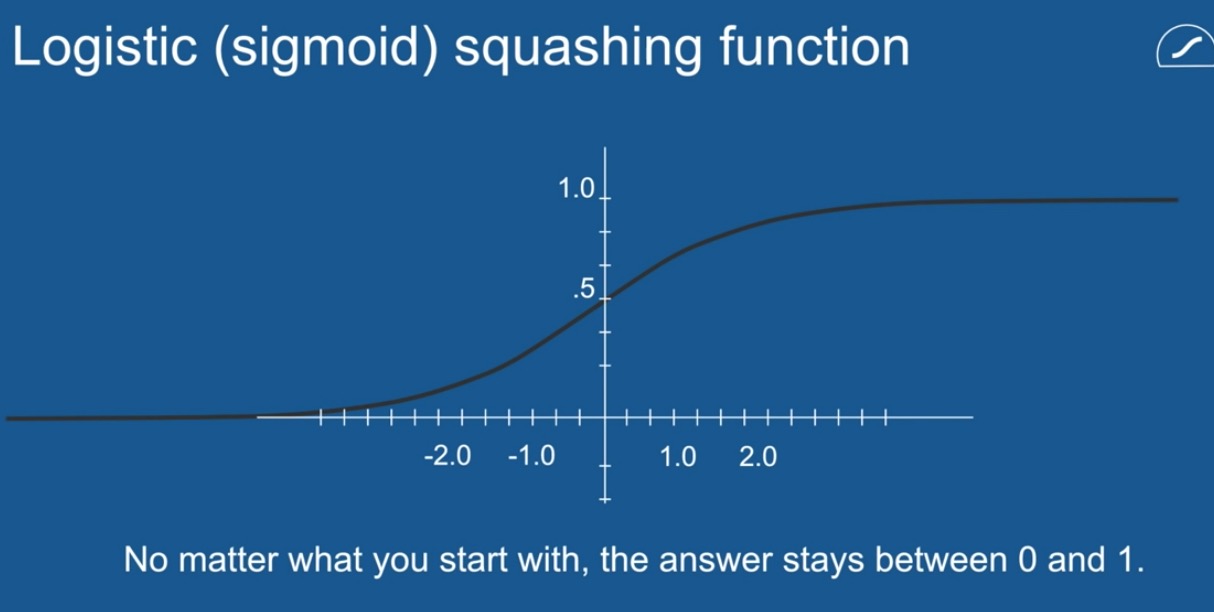
* computers native language – numbers
* 



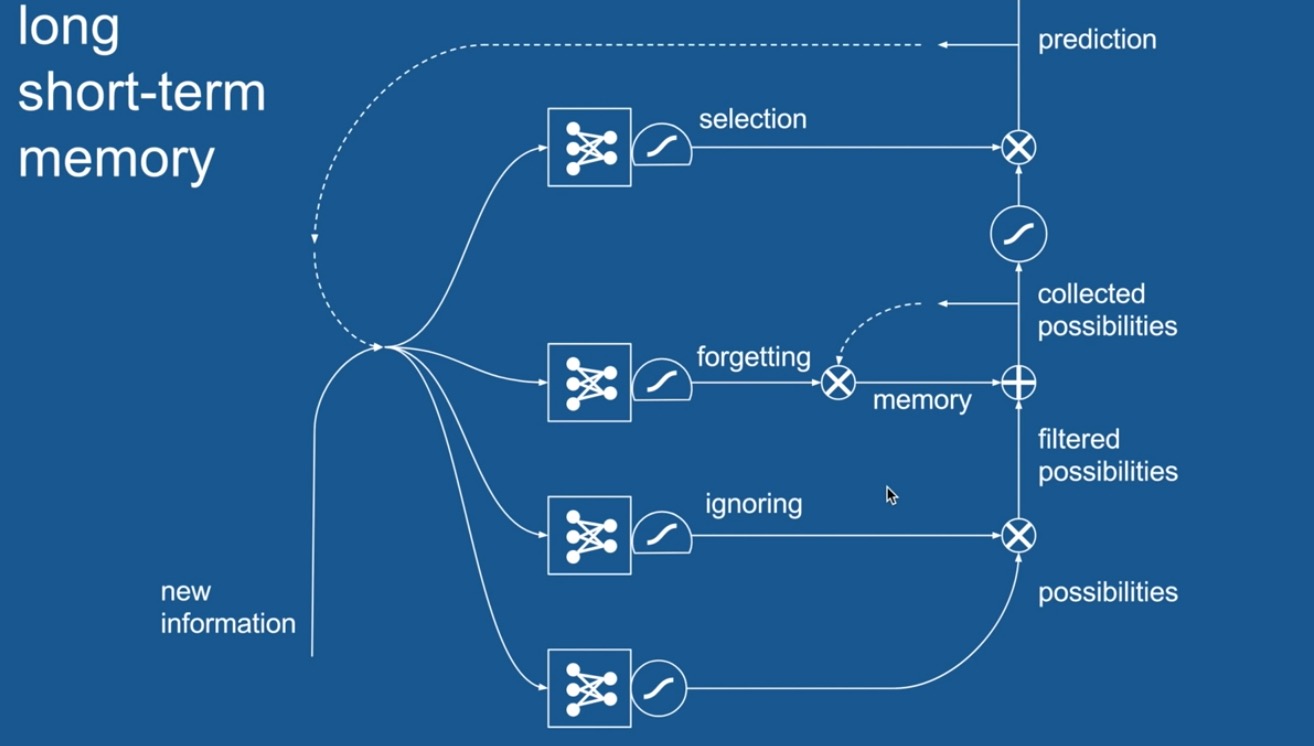
* in this example of how to write a good children’s book… the network is trained and when information passed for example is ‘saw’ or ’.’ there’s a high chance it will vote for a name to follow it as a name has to follow these so it will learn this. As these are the words in the dictionary that follow ‘saw’ and ‘.’ . Similarly if we predicted ‘saw’ or ‘.’ on previous timestep we expect those to vote also for a name



* this function that **occurs after the neural network**, converts scores to output between -1 and 1 - this means our loop wont explode if something gets multiplied by 2 each time as it always spits out -1<x<1



* another squashing function for gating – value always 0<x<1



* above shows LSTM in a diagram – has 4 neural networks – LSTM look back many timesteps