Long Short Term Memory (LSTM) the Way I Understand It

Caution! This discussion is going to be lengthy, twisted, brain teaser and probably brain numbing as well. Strong and frequent dose of coffee with dark chocolate will be in order.

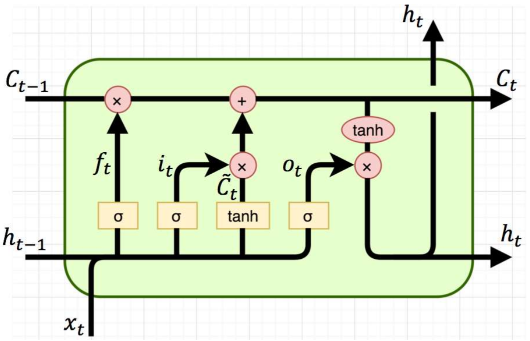
Given the limitation of RNN (https://www.linkedin.com/posts/mukeshrao1_recurrent-neural-network-as-a-state-machine-activity-6681424681699962880-iPNp) in learning long term dependencies, let us explore how LSTM (a variant of RNN) is able to overcome this limitation. Today, LSTM and its variant GRU, are the default choice for modelling sequential relations because they can learn long range dependencies.

The problem with RNN in learning long term dependencies appeared inform of long chain equations especially with respect to the internal states represented by matrix Wh. LSTM is able to overcome this problem by introducing two internal state vectors. One is short term internal state vector a.k.a. “Hidden State” and another one that act as long term state a.k.a. “Cell State”.

The “Cell State” and “Hidden State” influence each other and both are internal to the system. “Cell State” is derived from the interaction between “Hidden State” and the input trigger. Like RNN, every instance of the LSTM can emit an output as a function of the hidden state.

During the training process the output emitted at each instance can be used to estimate the error and use an error function with gradient descent to learn to represent the hidden state and internal state appropriately at each instance. Like in RNN, here too we have a set of weights which create the gradient vector fields where the cell state is mapped to point attractors in the vector field.

When we start exploring LSTM, we usually come across a picture (shown below) with various equations and flows representing a LSTM cell.



Pic 1

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

This diagram is by Christopher Olah (a respected figure in the world of deep neural network). In my view, this picture should be used as a summary at the end of the journey in learning LSTM. Using this to start the journey leaves one with more questions than answers. Where do these equations come from, what do they stand for, why those sigmoid and tanh operations are done… These questions bothered me too and I did not get any answers. The equations could not have come from thin air, there has to be some meaning behind all these and the lack of any convincing answer forced me to explore the field of RNN (shared in my earlier post) and associated technologies and thought processes. After more than a year and a half, I believe I have a much better understanding of RNN and LSTM. Thanks to the various authors of yester years whom I banked on. Will give reference to them at appropriate places in the post.

Since it is a common practice to introduce RNN and LSTM in the language processing domain, am going to do the same here. However, all these concepts apply in any domain that involves sequential relations between the inputs.

So, all set for the roller coaster?

It's a mind game after all!

If you play the game below sincerely, you will be using LSTM in your mind. To begin with, imagine you are engaged in a conversation with Ram, your fictitious colleague. It is a long conversation at the coffee machine. While he speaks, you mentally form a picture of what is he talking about, you try to get the context that he is referring to.

We all do this when we are listening. The meaning of a word /collection of words depend on the context in which it is used. So, we first have to form the context before we give a meaning to the word. For e.g. I take a train from home to work where I train people on AI. Here, the word “train” means locomotive in the context formed by (“take”, “from home”, “to work”). The same word means “to teach” when the context is formed by (“people”, “AI”). So, unless I formulate the context, assigning meaning to a word is not possible. The word “Train” has no particular meaning. It depends on the context.

With this understanding, let us put the interaction between you and Ram on a slow mode and you have to rationally guess the context and change the context if required to get the whole meaning of the conversation –

Step	Ram	You guess the possible context in your mind
1	“While working from home has its own advantages…”	He is going to compare pros and cons of W.F.H. and N.W.F.H **Note:** W.F.H. stands for Working From Home. N.W.F.H. stands for not working from home. W.F.H sounds similar to something else…. Hence this note ☺
2	It is a new normal given the lockdown…	Oh! He is not comparing W.F.H Vs N.W.F.H., he is going to talk of the lockdown we are in due to Corona virus
3	Which is the only way to slowdown Corona spread	Ok, so it is all about the Corona and its impact on work life… what a bore!

4	Which has had a huge negative impact on the GDP of all nations…	Oh god! He is going to discuss economy and finance.. not my cup of tea… wonder how long this is going to take
5	Some of which are at war…	Impact of war on the economy… am sure he is going to rattle off. I wish I was in the loo!
6	With the invading swarms of locust	Oh! He is talking of the war on the locust, I thought he was talking about the tensions in Ladakh and SCN….
7	That can significantly damage the agriculture output…	Hmmm… he is talking about impact of locust attack on agriculture… what a confused man, going all over the place and not focusing on one topic.
8	…	Ram, I need to go… I have a meeting to attend (you mean better spend time on Mukesh Rao's post on LSTM ☺)

Pic 2

When you get the statement in parts, as in the sequence above and you are not aware of what the next piece is going to be, you mentally form a context which you believe is the background for the discussion. For e.g. "While working from home has its own advantages"… may lead you to believe the discussion is about work from home Vs working in office. However, at the next point, the words "It is the new normal given the lockdown", nudges you to change the context to the lockdown related developments because the new words appear more frequently in the topics related to corona virus than in the context of W.F.H. and its pros and cons.

The point 3 further reinforces our belief that the context is Corona virus pandemic. However, Point 4 once again nudges you to change the context to the geopolitical tensions while at Point 5 you change the context from the geopolitical tensions to a new context of the invading locust and impact on agriculture.

This is a very common experience when we are involved in any conversation. Many of us would have experienced this. I call it the mental context switch. This is akin to the internal state transitions in the state machine we discussed in RNN. Let us explore this a bit more.

Mental Context Switch in Detail

1. At the starting of the game you did not have any idea of what it is going to be. Probably you were guessing what it could be given we are discussing LSTM. Whatever it was, let that be your initial mental state.
2. When you got the first set of words "While working from home has its own advantages…" you took the key words from this (W.F.H. , advantages)

3. Internally you calculated the probability of the initial mental context given the key words and you decided that the initial context is not relevant and hence needs to be changed
4. You changed the context to W.F.H. Vs N.W.F.H pros and cons because the probability of this new context given the key words is relatively high

(Caution: This is going to be slippery…).

   a. We assign meaning to the new words based on the context we think they are being used in. For e.g. the word 'War' in the context of the geopolitical tensions means something different than in the context of "swarm of locust"
   b. When we change mental context given the new word, we usually carry forward some information from previous context to the next.
   c. The information content carried forward may be zero or greater. For e.g. when we changed from step 1 to step 2 in the game, we carried forward some information about W.F.H. Vs N.W.F.H context to the new context formed by lockdown, corona

5. As per points 4a and 4b together, the two (new context and the word) strongly influence each other, they give meaning to one another.
6. Together they make conversation meaningful as they mutually convey some information

Given these steps that we implicitly take during a mental context switch, it is paramount to understand what a context is, what is a topic and the interaction between context and a word.

What is a Context?

Context means background, a frame of reference that gives meaning to the whole conversation. It consists of a mixture of topics (will be relatively easy to understand if you are aware of the problem of topic identification given a document, if not, don't worry… enjoy the roller coaster…). The mixture contains topics in various proportions. For e.g. in the game above, the topics were W.F.H. culture, Corona virus, Lockdown, War, Locust, Agriculture, crop damage, economy etc. From these topics, we mentally formed different contexts by assigning different weights to each topic in defining a context.

When the context was geopolitical tension, the topics that helped form this context were corona, lockdown, war etc. But when the key word "locust" occurred, the context changed to "impact of locust attack on agriculture" for which the topics corona, lockdown were not so important anymore. More important topics were "agriculture", "crop damage" etc.

Thus, to change the context, we change the degree of importance assigned to the various topics implicitly referred to in the conversation. So, if we take topics to be from T1 to Tn, we can form contexts C1 to Cm using these topics in various proportions, though not all Cm may be meaningful.

Topics / Context	C1	C2	…	Cm
T1	50	30	10	55
T2	40	30	50	30
Tn	10	40	40	15
Total	100	100	100	100

Pic 3

Note: Numbers in the grid are only to explain the idea of importance of topic in a context.

This is not new, those of you who have worked on recommendation systems using SVD (Singular Value Decomposition) techniques would be able to see the connection. SVD is a matrix decomposition technique that takes as input a matrix such as customers and movies ratings. When this matrix is decomposed using SVD, it results in new matrices of which one acts like a mapping of customers to some hidden attributes and the other matrix maps movies to those hidden attributes. The hidden attributes are equivalent to the implicit topics in this discussion. The hidden attributes in various proportions characterize a movie (context) and a customer who relates to that context more, rates the movie high.

Ref: https://developers.google.com/machine-learning/recommendation/collaborative/matrix

Topics

Topics are the implicit subject matter of a discussion (verbal or written). A conversation can cover a range of topics. For e.g. the following online conversation between a doctor and a person taken from https://timesofindia.indiatimes.com/life-style/health-fitness/health-news/doctor-speaks-most-commonly-asked-questions/articleshow/74695387.cms

I must tell you today that this virus is not a deadly virus. It is a flu virus. The only problem with this virus is, it is highly infectious, contagious and the infectivity rate is ten times more than a normal flu virus. That is the only matter of concern. If you see the mortality, the mortality is only 2.5 to 3 per cent, that too in elderly patients, beyond 60 or 70. And it is only in those patients who have underlying co-morbidity like hypertension or chronic heart disease or diabetes or cancer or they are immunologically compromised. So, I don't think everybody is vulnerable. 80-85% of patients have very minor infection. They recover. They just need quarantine. They need isolation. They need rest at home.

The topics include virus, flu, elderly patients, co-morbidity, and quarantine. Topics are basically subjects upon which one can converse independently of others. For e.g. one can have a conversation on normal flu completely independent of another conversation on quarantine.

A mixture of topics consists of various topics with different degrees of focus in a conversation. For e.g. in the discussion above, the common flu was given a passing reference while the co-morbid conditions got relatively higher coverage.

If we take all the conversations on Corona virus and remove the stop-words (words used for syntactical correctness which do not carry any subject information for e.g. "The", "it" etc.) and take a frequency count of all the words (key words), we will notice that some words occur more frequently than others. For e.g. "Pandemic", "isolation", "Quarantine", "Comorbidity", "Social distancing" etc. These words may not occur so frequently when we are discussing common cold.

The conversation started with a line "*I must tell you today that this virus is not a deadly virus. It is a flu virus."* Would lead one to believe it is a discussion of the flu virus. That is the context. But when subsequent line "*The only problem with this virus is, it is highly infectious, contagious and the infectivity rate is ten times more than a normal flu virus."* occurs, we realize that it is not flu but not sure what it is. Then come more words "*And it is only in those patients who have underlying co-morbidity like hypertension or chronic heart disease or diabetes or cancer or they are immunologically compromised."* Given the key words, it is likely the discussion is about Corona virus. Hence we switched the context.

If we had a statement such as -

"I must tell you today that this virus is not a deadly virus. It is a flu virus. Patients who have underlying co-morbidity like hypertension or chronic heart disease or diabetes or cancer or they are immunologically compromised are more likely to develop serious conditions".

The context here is flu virus and the subsequent key words do not appear so frequently in this context. Though the statement is syntactically and semantically accurate, it is going against the grain! It is not in line with what is already known. It is causing more confusion than give any useful information. Words such as co-morbidity do not occur when context is common flu. Which means, when there words are used, the context is not common flu and when context is common flu, these words do not occur.

Key words help set the context and the context with the key words enrich us with some useful information for e.g. Corona virus can become dangerous for people with underlying co-morbid conditions.

The relation between key words and context can be expressed in terms of joint probability and how strongly they support each other to convey useful information. Joint probability is how frequently they occur together and this is intuitive but what should the metric be to represent the mutual support or mutual contribution to meaningful information?

Let us call this metric MI (Mutual Information metric). Along with JP (Joint Probability) let us fix a scale for these two metrics. All of us know that JP can range from 0 to +1 (valid range for probability measurement).

For fixing the valid range for MI we need some discussion. But we can use following situations to help us –

Joint Prob / Information content	Yes	No
High	MI should be large +ive For e.g. co-morbidity in context of Covid. (Q1)	MI should be zero. For e.g. virus in context of Covid (Q2)
Low	MI should be –ive For e.g. Kidney damage and Covid context (Q3)	MI should be zero. For e.g. Coffee and Covid context (Q4)

Pic 4

1. Word and the context have high joint probability and convey useful information (Q1), MI should be +. For e.g. Co-morbidity in the context of Corona Virus Pandemic. The positive MI indicates high joint probability and high information content
2. Word and context have high joint probability, convey no information (Q2), MI should be 0. For e.g. Virus in the context of Corona Virus Pandemic. Dropping the word will not lead to loss of information
3. Word and context have low joint probability, convey useful information (Q3), MI should be –ive. For e.g. Kidney failure in context of Covid
4. Word and context have low joint probability and no useful information (Q4), MI should be 0. For e.g. Coffee in the context of Covid.

The reason we wish to differentiate between Q1 and Q3 where both cases we get useful information is to reflect the fact that in one case the joint probability is high while in other it is low.

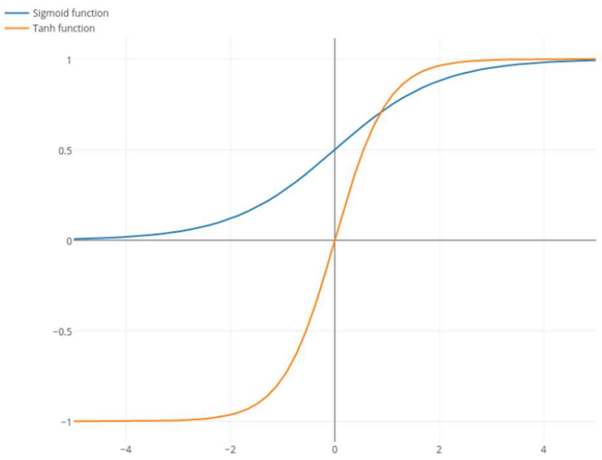
If we do not do this then cases where the word rarely occurs in a context but conveys useful information will not stand out. For e.g. “kidney failure” in the context of Covid. This word is not as common as “dry cough” in the same context. Both convey useful information.

This situation cannot be clubbed with quadrant 1 (because of difference in joint probability) or quadrant 2, 4 (because of difference in useful information content). How do we represent this concept of joint probability and information contribution keeping all the four cases in view? Whatever the way we do it, it should capture the information when available otherwise ignore that is give results in Q1 and Q3 and ignore Q2 and Q4.

This is where the mathematical function of Sigmoid, Tanh and their product come in handy. Let us understand these two functions and check what happens when we take product of these two functions.

Properties of Sigmoid and Tanh functions

Sigmoid and Tanh functions in two dimensions (input X and output Y) look like



Pic 5

1. Input to sigmoid and tanh range from –infinity to + infinity
2. Both are continuous and smooth functions
3. Both are asymptote
4. Sigmoid
    a. Output ranges from 0 at input x = –infinity to 1 at input x = + infinity
    b. Is .5 at input x = 0
    c. Can be used to scale data and represent probability
5. Tanh
    a. Output ranges from -1 at input x = -infinity to 1 at input = +infinity.
    b. Is 0 at input x = 0
    c. Can be used for to represent correlation between entities (https://mathematicsinindustry.springeropen.com/articles/10.1186/s13362-016-0018-4).
    d. Correlation is a measure of association, mutual information between entities (Ref: https://en.wikipedia.org/wiki/Mutual_information).

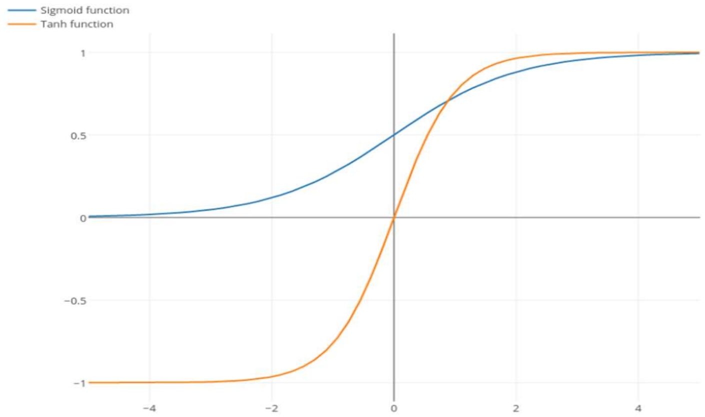
Thus, we can use Sigmoid to calculate probability and also scale outputs between 0 and 1 while Tanh function can be used to calculate correlation and also for scaling outputs between -1 and 1.

Given the properties of Sigmoid and Tanh, let us explore how we can use these functions to extract useful information from a context given a word and carry forward the information to define the new context.

“Carry forward the information to the new context” may sound a bit strange. We can understand carry forward of say credit, stock at hand etc. Can we measure information? No doubt useful information enlightens, gives clarity but how do we measure information. How do we quantify it?

Mutual Information

Let X represent some measure of the interaction between a word and its context (both represented as vectors in the feature space).

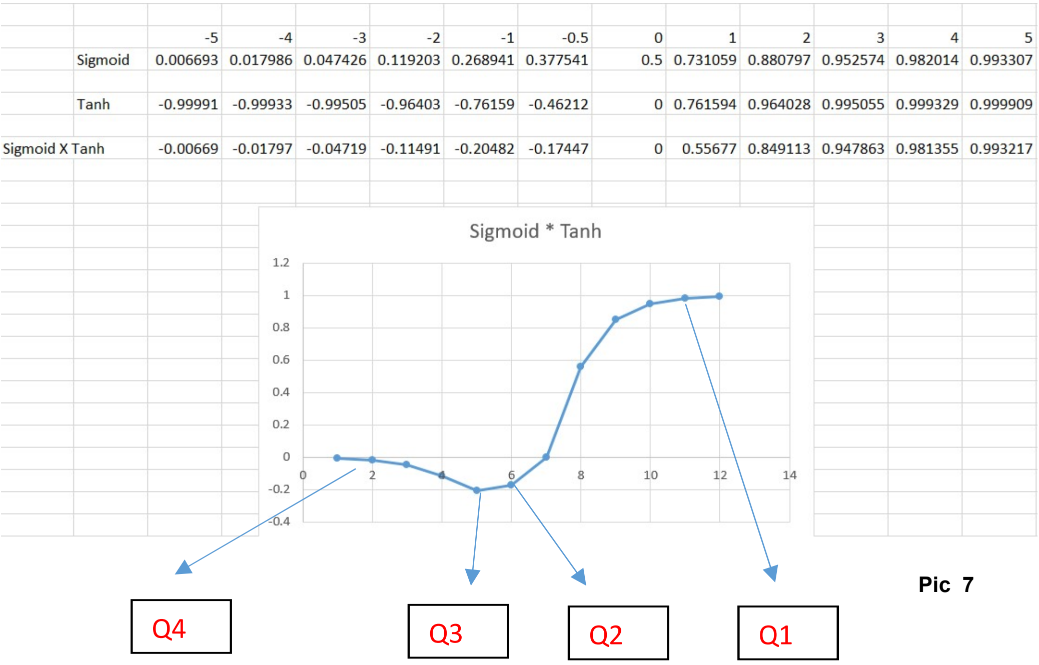


Pic 6

1. When input X is large positive number
    a. Probability (output of Sigmoid) is close to 1
    b. Correlation(output of Tanh) is close to 1
    c. Sig * Tanh is close to 1
    d. Larger the magnitude of the product reflect the two entities have a strong positive influence on one another. This is the Q1 in grid pic 4
2. When input X = 0
    a. Probability is .5 (uncertainty is high), correlation is 0
    b. Two vectors with probability of .5 indicate they may or may not occur together
    c. Correlation (output of Tanh) will be 0 indicating no influence between the two vectors
    d. Thus product of Sigmoid and Tanh when 0 indicates the input word and context are independent and zero influence of one on the other. This is Q2 in pic 4

3. When input X is slightly <0
    a. probability is <.5 (a low chance of the two occurring)
    b. Negative correlation of a small magnitude
    c. The product of Sigmoid and Tanh is a small negative value indicating low probability and when they occur, this will be a rare but possible case. This is Q3 in pic 4
4. When input X is significantly less than 0
    a. correlation is close to -1
    b. Sigmoid * Tanh is a –ve fraction (max magnitude =.2)
    c. If two entities have very low probability close to 0, they will have large –ve correlation indicating that the word and context cannot occur together. This is Q4 in pic 4

These points are summarized in the diagram below –



		-5	-4	-3	-2	-1	-0.5	0	1	2	3	4	5
	Sigmoid	0.006693	0.017986	0.047426	0.119203	0.268941	0.377541	0.5	0.731059	0.880797	0.952574	0.982014	0.993307
	Tanh	-0.99991	-0.99933	-0.99505	-0.96403	-0.76159	-0.46212	0	0.761594	0.964028	0.995055	0.999329	0.999909
Sigmoid X Tanh		-0.00669	-0.01797	-0.04719	-0.11491	-0.20482	-0.17447	0	0.55677	0.849113	0.947863	0.981355	0.993217

Pic 7

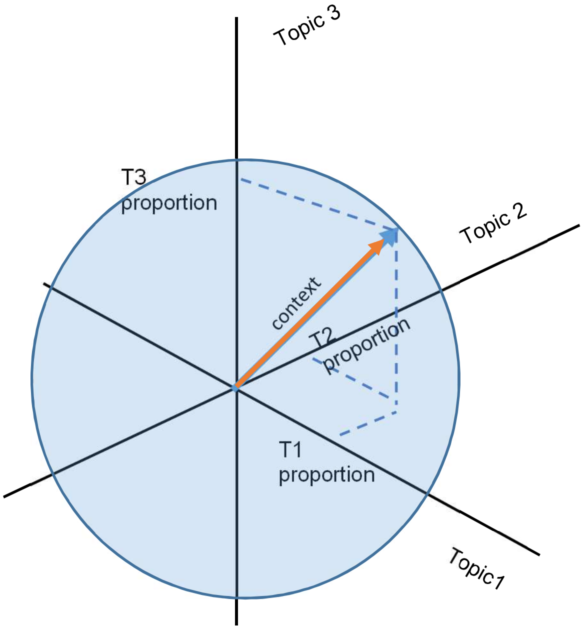
Thus Sigmoid and Tanh functions can be used to capture the relation between a word and the context and the mutual information they carry to define next context. Now that we have mathematical functions to help us, let us step into the virtual world of maths to represent all the concepts discussed till now. Representing the real world in a virtual mathematical world is an old idea since the days of ancient Indian thinkers and Greek philosophers.

Mathematical Representation of Real World

With the discussion on context, context being a mixture of topics in varying proportions, correlation and probability between context and words resulting in mutual information, context switch carrying forward mutual information etc. I have only one more point to discuss before we get into LSTM. It's about the virtual reality ☺.

All models are built in a virtual world of feature space. We talk of models as some best fit line or boundary or a manifold/surface or even hyper surface in hyper space. Nothing of this sort really happens. Whatever happens, happens only in the given data. This could be a mathematical formula connecting target variable to independent variable (y_pred = mX + C), likelihood ratios, association rules etc. However, we still resort to the virtual feature space when we try to understand what a model is, as it is very intuitive and helpful in understanding and explaining. This is exactly what ancient wise people from India and Greece used to do. Represent the real world using Geometry and Mathematics to understand and explain all the reality around. A web search on this topic will reveal a ton of examples. (Ref : https://en.wikipedia.org/wiki/History_of_geometry).

LSTM too is a mathematical representation what happens when we process sequential natured data in the real world. Especially when we process languages. This representation begins by visualizing the virtual feature space shown below.



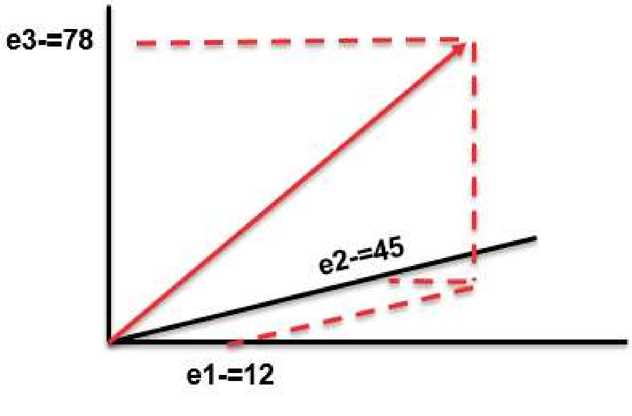
Pic 8

1. Each dimension is a latent topic. The value of each dimension ranges from 0 to 1
2. A collection of topics in various proportions between 0 and 1 (including the limits) is a context
3. The current context can be represented as a vector emanating from the origin
4. The dashed lines indicate the proportion of each topic that go to make the context
5. Imagine a sphere of unit radius around the origin. Every point on the sphere is a context. Not all may be relevant

6. LSTM employs a hidden state vector and a cell state vector. To begin with, both are same. Cell state (C) is orange and Hidden state (H) is blue
7. Every trigger / new word is also represented as a vector in the same feature space. Word vector is not shown above. This is discussed in detail in next section.
8. The H vector is used with every new word to do the probability and correlation calculations and the results are used to update the C vector
9. Once the C vector is updated, it is used to update the H vector itself
10. The H vector updates based on inputs is what is the Short Term Memory while the update to C vector is Long Term Memory
11. This concept of using short term memory to update long term memory is similar to Associative Memory in biology. Ref: *Short-Term Memory Emmanuel Guigon and Yves Burnod / The Handbook Of Brain Theory and Network , Classical Learning Theory and Neural Networks Bennet B Murdoc / The Handbook Of Brain Theory and Network*
12. Everything (Hidden State vector, Cell State vector, Trigger) in this virtual space is defined and represented as a vector
13. The H vector will range from 0 to 1 in magnitude while the C vector can take any value. The sphere around the origin is scope of the H vector. Why is this so? This will become clear when we look at some more details

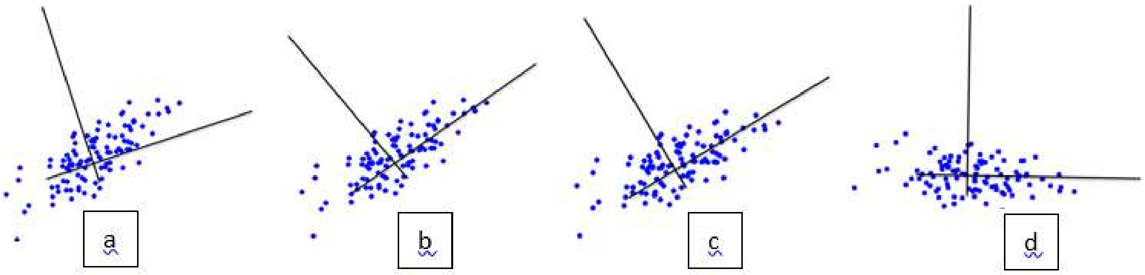
Vectorised Inputs/Triggers

The inputs are represented as vectors as we saw in the case of RNN. Every input is expressed as a set of values taken by each dimension (that represents a latent topic) in defining it. These values could be considered as relevance of the word given the dimension. The number of elements to define an input is the same as the number of dimensions used to define the LSTM feature space. Both the state vectors and input exist in the same feature space. For e.g. if input word “The” is expressed as a vector of 3 elements such as [12, 45, 78], it can be represented in the same three dimensional feature space as a point as shown below.

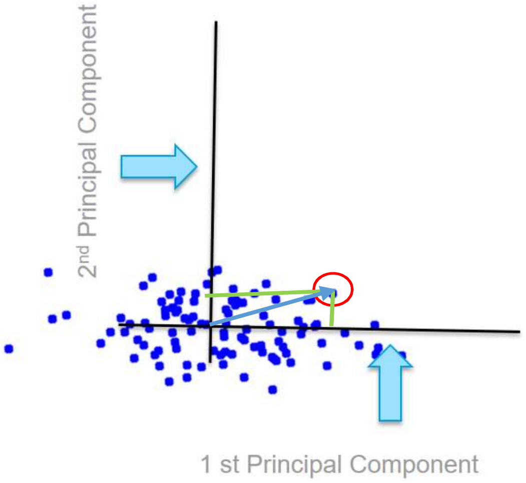


Pic 9

If the concept of dimensions being topics is still proving to be a brain teaser, think of what Principal Component Analysis (PCA) does for instance. Given the distribution of data (blue points), PCA looks for directions of max spread in the original feature space and use those directions as new coordinates (picture 6d) below. What are these directions? These directions can be considered as some latent topics which together represent all the information in the original feature space. Information that exists in form of variance and covariance.



Every data point in the feature space is mapped to the principal components and is represented in terms of the components (diagram d). That means every data point now is represented by the topics in terms of importance of the word given the topics.

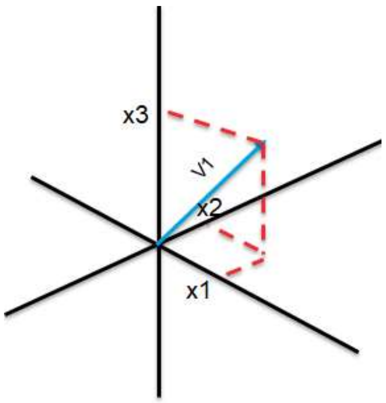


Pic 10

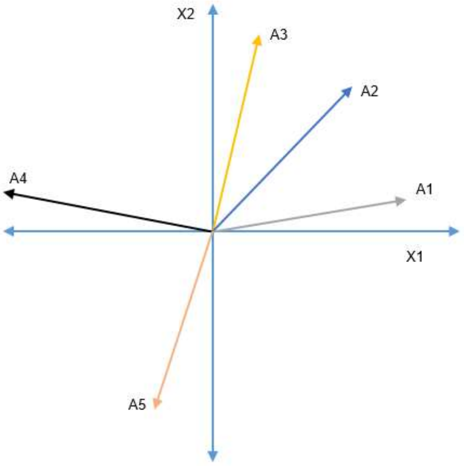
The point circled in the pic above is now a vector defined in terms of principal components and the principal components are latent topics

Modifying the orientation of a vector in feature space

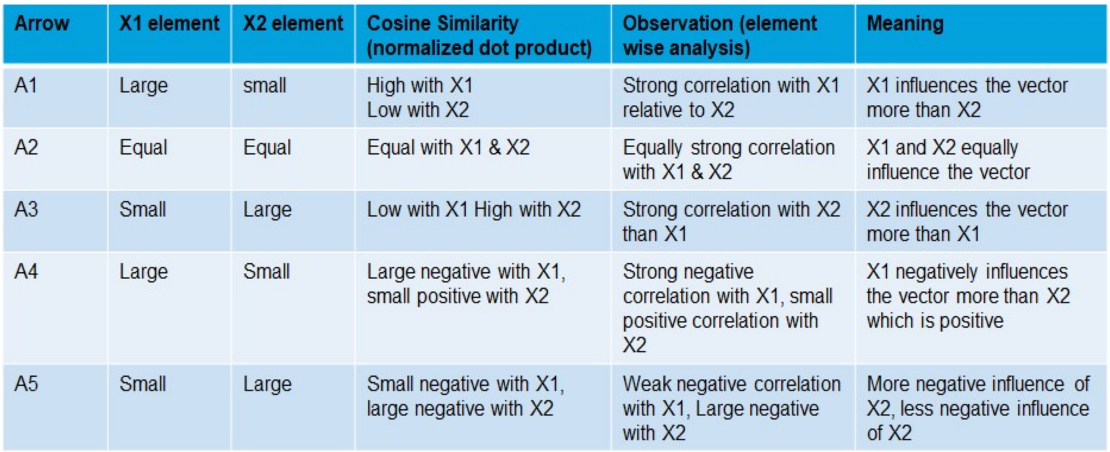
The orientation of a vector in the feature space represents relative importance of the word given the topic. To re-orient a vector, the element values [X1, X2, X3] need to change i.e. the relevance of the word for each topic need to change.



The more relevant a word is for a topic, the closer the vector will be to that topic dimension than others. If all the dimensions have equal influence, the vector will be at equal angular distance from all the topic dimensions.



Pic 11



Arrow	X1 element	X2 element	Cosine Similarity (normalized dot product)	Observation (element wise analysis)	Meaning
A1	Large	small	High with X1 Low with X2	Strong correlation with X1 relative to X2	X1 influences the vector more than X2
A2	Equal	Equal	Equal with X1 & X2	Equally strong correlation with X1 & X2	X1 and X2 equally influence the vector
A3	Small	Large	Low with X1 High with X2	Strong correlation with X2 than X1	X2 influences the vector more than X1
A4	Large	Small	Large negative with X1, small positive with X2	Strong negative correlation with X1, small positive correlation with X2	X1 negatively influences the vector more than X2 which is positive
A5	Small	Large	Small negative with X1, large negative with X2	Weak negative correlation with X1, Large negative with X2	More negative influence of X2, less negative influence of X2

Pic 12

Given all the discussion we had till now, I suggest you summarize the points. From the next section onwards we will get into the core part of LSTM. When we do that, we need to connect the dots (concepts discussed) to get the idea behind LSTM cell (the way I understood it). For your convenience –

1. Context Switch
2. Context
3. Topics
4. Feature space with topics as dimensions
5. Vector representation of context and word
6. Sigmoid and Tanh function properties
7. Sigmoid for probability and scaling, Tanh for correlation and scaling
8. A word and context when strongly correlate, convey useful information
9. Multiplying Sigmoid and Tanh helps extract useful information from a word context combination
10. Both Hidden State and Cell State are represented as vectors in the feature space. The input word / trigger is also represented as a vector in the same space
11. The orientation of a word vector when factored into individual dimensions indicate the relevance of the vector against each of the latent topic that represents a dimensions
12. The close a word vector is to a dimension, the more the relevance of the word in that topic
13. The orientation of a context vector when factored into individual dimensions indicate the proportion of each topic contribution in defining the context
14. The H vector can at the max have a magnitude of 1 while the C vector can have any magnitude

We will connect all these points to understand how LSTM cell works.

Short Term and Long Term Memory

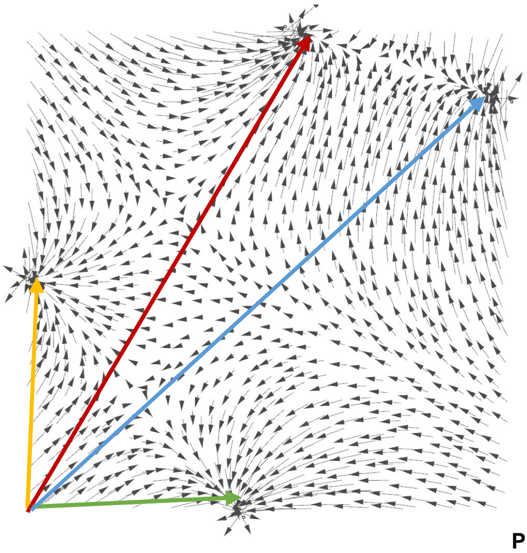
Many of us remember our childhood friends vividly. We can easily recall those incidents that we enjoyed together, we can even recall their face clearly. At the same time we also tend to forget what we wore last Monday for instance. Through experiments, it has been demonstrated in Neurology that our learning process involves two steps where the lessons are first stored in short term memory and from there it gets etched into long term memory. Both memories fade with time but short term memory fades relatively faster. No wonder, practice makes one perfect. Perfection is that state where things happen automatically without much effort (long term memory), while practise is the process of transferring the short term memory (immediate lessons learnt) into long term memory. Refer*: "Short-Term Memory" by Emmanuel Guigon and Yves Burnod / The Handbook of Brain Theory and Neural Networks.*

In RNN/LSTM context, the Long Term Memory refers to the mapping of internal states to the attractors in the vector field (discussed in RNN). It also includes the ability to transition from one state to next on occurrence of a trigger i.e. learn the transition trajectories. The LTM is represented by the Cell State vector C.

Short Term Memory (STM) refers to the interaction between the hidden state vector H and the input trigger that is used during the training stage to update the Cell State vector C i.e. the LTM.

Vectors for Short Term and Long Term Memory

As we discussed in case of RNN, any dynamic system can be understood in terms of its internal states were the internal states can be represented in terms of state variables (known as topics in this discussion) and each valid state is represented by an attractor in the state space. Each attractor is a vector. Same is the case in LSTM though, the mechanism of constructing the vector fields with attractors is slightly different.



Pic 13

In pic 13 we see a sample of the gradient vectors, converging into various state attractors which can be represented by a vector and each vector thus represents a valid internal state. The flow of the gradients represent the state transition trajectory. This is the result of a completed training phase of an LSTM (similar to RNN) based system. Let us see how this can be achieved in LSTM.

LSTM Mechanism

Let us step back and restart at the point where we have an empty feature space made of three dimensions (limited to three due to our visualization limitations). The individual dimensions represent latent topics.

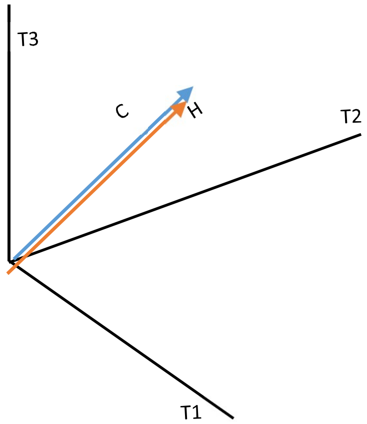
Given this feature space of three dimensions, recall the game of guessing you played… repeated here for ease of reference.

Step	Ram	You guess the possible context in your mind
1	"While working from home has its own advantages…"	He is going to compare pros and cons of W.F.H. and N.W.F.H **Note:** W.F.H. stands for Working From Home. N.W.F.H. stands for not working from home. W.F.H sounds similar to something else…. Hence this note ☺
2	It is a new normal given the lockdown…	Oh! He is not comparing W.F.H Vs N.W.F.H., he is going to talk of the lockdown we are in due to Corona virus
3	Which is the only way to slowdown Corona spread	Ok, so it is all about the Corona and its impact on work life… what a bore!
4	Which has had a huge negative impact on the GDP of all nations…	Oh god! He is going to discuss economy and finance.. not my cup of tea… wonder how long this is going to take
5	Some of which are at war…	Impact of war on the economy… am sure he is going to rattle off. I wish I was in the loo!
6	With the invading swarms of locust	Oh! He is talking of the war on the locust, I thought he was talking about the tensions in Ladakh and SCN….
7	That can significantly damage the agriculture output…	Hmmm… he is talking about impact of locust attack on agriculture… what a confused man, going all over the place and not focusing on one topic.
8	…	Ram, I need to go… I have a meeting to attend (you mean better spend time on Mukesh Rao's post on LSTM ☺)

Pic 14

The mental context switch is akin to internal state transition in a state machine. Where the words act like triggers nudging a change in the mental context (which is akin to internal state change in state machine).

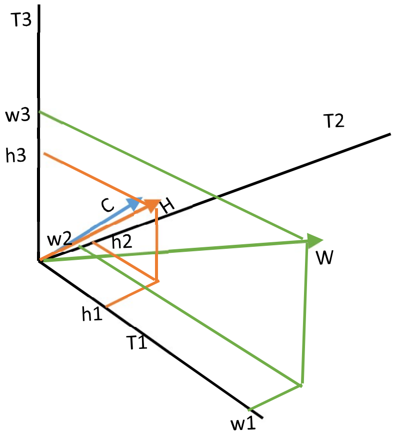
Let us try to give a visual representation to this whole discussion. Let there be two vectors called 'C' and 'H' pointing in the same but random directions in the feature space made of three dimensions (Ref pic 15). The vector C is Cell State vector and points in the direction of current context while H is a hidden vector which is used to reorient the C vector on context change.



Pic 15

I have separated them a bit to show both of them. They are pointing in some arbitrary direction representing some arbitrary context.

A new word (trigger) 'W' occur in the form of a vector of same three dimensions, shown below as green arrow. The orientation of the word is dependent on the relevance of the word for the different topics (refer to fig 16 and fig 17).



Pic 16

Given the word, how relevant the current context C is. For this, take the H vector (which represents C), break it into its components (h1, h2, h3), and take the components of W (w1, w2, w3) calculate the **Sigmoid (H, W)** elementwise. Let H vector be initialized to [1, 1, 1]. Suppose this results in the following -

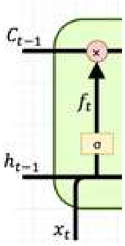
Sigmoid(h1+w1 , h2+w2, h3+w3) = [.70 , .45, .60]

This can be interpreted as .70 of T1, .45 of T2 and .60 of T3 mixture of topics will go into defining the next context or Cell State Cnew. To define the Cnew, first multiply current C with Sigmoid output elementwise as shown below.

1. Let C = [1, 1, 1] (initialized same as H)
2. C_{limbo} = C1 * .70 , C2*.45 , C3*.60 = [.70, .45, .60].

The reason this is Climbo is, we have yet to carry forward mutual information to define the Cnew.

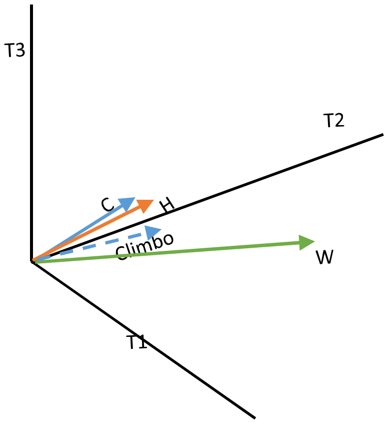
What we have achieved in the two steps above, is implementing the "**Forget Gate**". From the given C we have taken a certain fraction of the topics leaving rest out (forgetting).



Pic 17

To update C_{limbo} to C_{new} we need to carry the mutual information given the word and current context h. For this we will use Tanh function elementwise. The output of Tanh multiplied with the output of Sigmoid helps us extract the information (Refer: section *"Mutual Information"*).

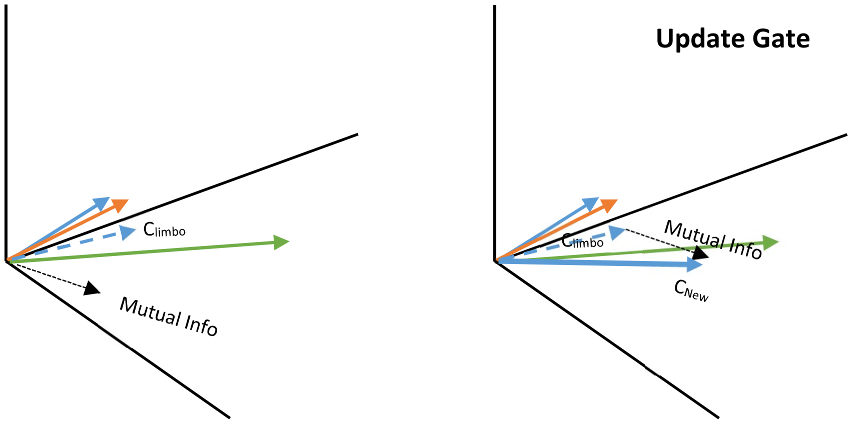
1. Tanh(h1+w1 , h2+w2, h3+w3) = [.75 , .40, .55]
2. Sigmoid(h1+w1 , h2+w2, h3+w3) = [.70 , .45, .60]
3. Mutual information from word W and current context H is -
   a. Mutual information = Tanh * Sigmoid elementwise = [0.52, 0.18, 0.33]
4. Cnew = Climbo vector + Mutual Information vector (this is a simple vector addition as shown below)



Forget Gate

Pic 18

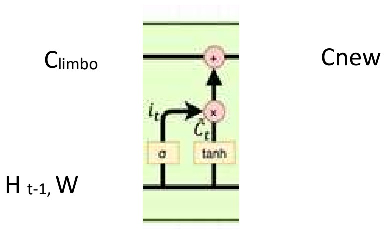
In the picture above we use the forget gate to take relevant portions of the current context creating a C_{limbo} Cell State. Next we calculate the mutual information which too is represented as a vector (dashed black).



Pic 19

This is added to the C_{limbo} state vector (Pic 19) thru simple vector translation and addition to get the new Cell State C_{new} (the new blue line)

This is what is achieved by the **"Update Gate"**

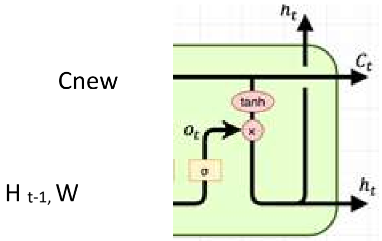


Pic 20

Now that the Cell State is updated, we need to represent this new Cell State in the Hidden state H i.e. update the H vector too. This is done in two steps –

1. Take the relevant part of the H state as we did earlier to get the C_{limbo}
2. Take Tanh of the new Cell State vector. The reason for taking Tanh of C_{new} is that the addition (+) operation can make the magnitude of H vector quite large leading to exploding and vanishing gradient problems
3. Multiply the Sigmoid elementwise with Tanh of Cnew to get the H_{t+1}

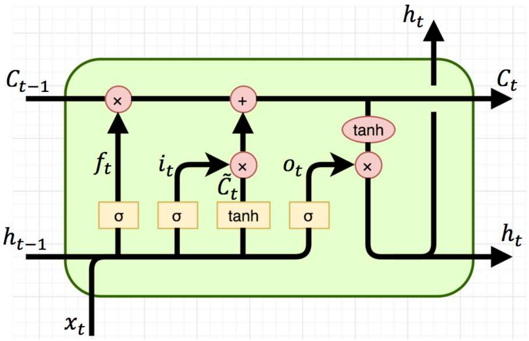
This is achieved by the "**Output Gate**"



Pic 21

LSTM Cell in Summary

Assuming you were able to connect the dots, now when you look at Colah's LSTM Cell, you will notice it is a beautiful summary of the entire discussion in a pictorial format.



Pic 22

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

To summarise, LSTM is a mathematical model of the process that we all adopt in any interaction to gain information/ knowledge.

1. We start by holding some idea as a context
2. With every input entering in a sequence, we automatically evaluate the relevance of the context given the input
3. If the context is not relevant, we modify the context based on the input. That context is relevant which along with the word conveys useful information

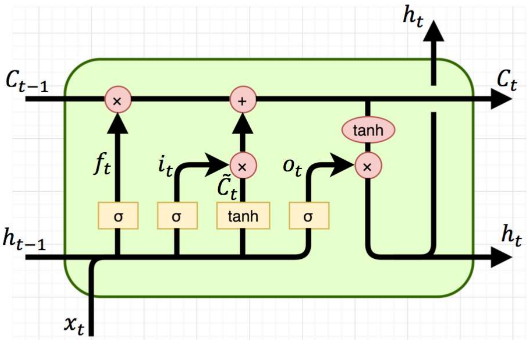
How about the weights and bias?

I have not referred to weights and bias at any step. I assumed all the weights are same and unit and so is bias. The reason for keeping it out of discussion till now is to focus on the concepts i.e. definition of the gates in particular and the LSTM Cell on the whole. With that behind us, now we can introduce weights for the input vector W, the H vector and C vector. These weights will be associated with each of the gates. Weights act as knobs to fine tune the whole setup and in the process get a generic model that will learn not specific sequential pattern but overall a generic set of states and state transitions.

The weights associated with the gates (Wf, Wi, Wc, Wo) is what creates the vector field with state attractors and the trajectories during the learning process. Let us now explore the learning process. We need to look at the chain equation. This will help us understand how the vanishing gradients (which plagued the RNN) are avoided in LSTM.

Training A LSTM Cell

The Ht generated by the output gate is used to generate output at every instance of the LSTM Cell. This appears as the Ht going vertically out of the LSTM cell. Thus the error gradient of a single instance LSTM cell will flow downwards from this point onwards.



Pic 23

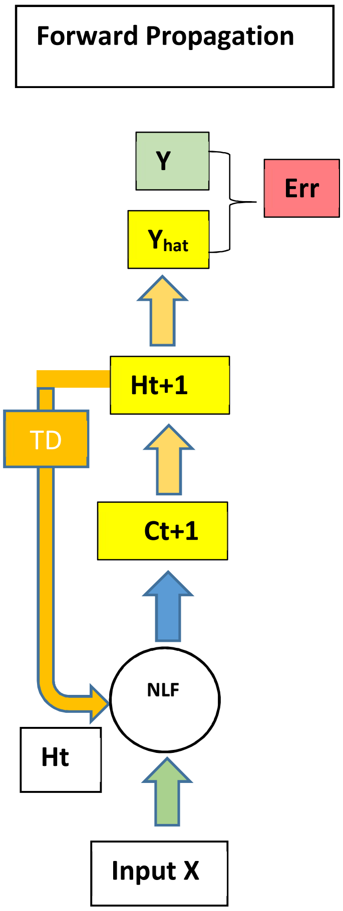
Ref: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

To understand the error gradient back-prop let us list down the equations for the gates and intermediate calculations.

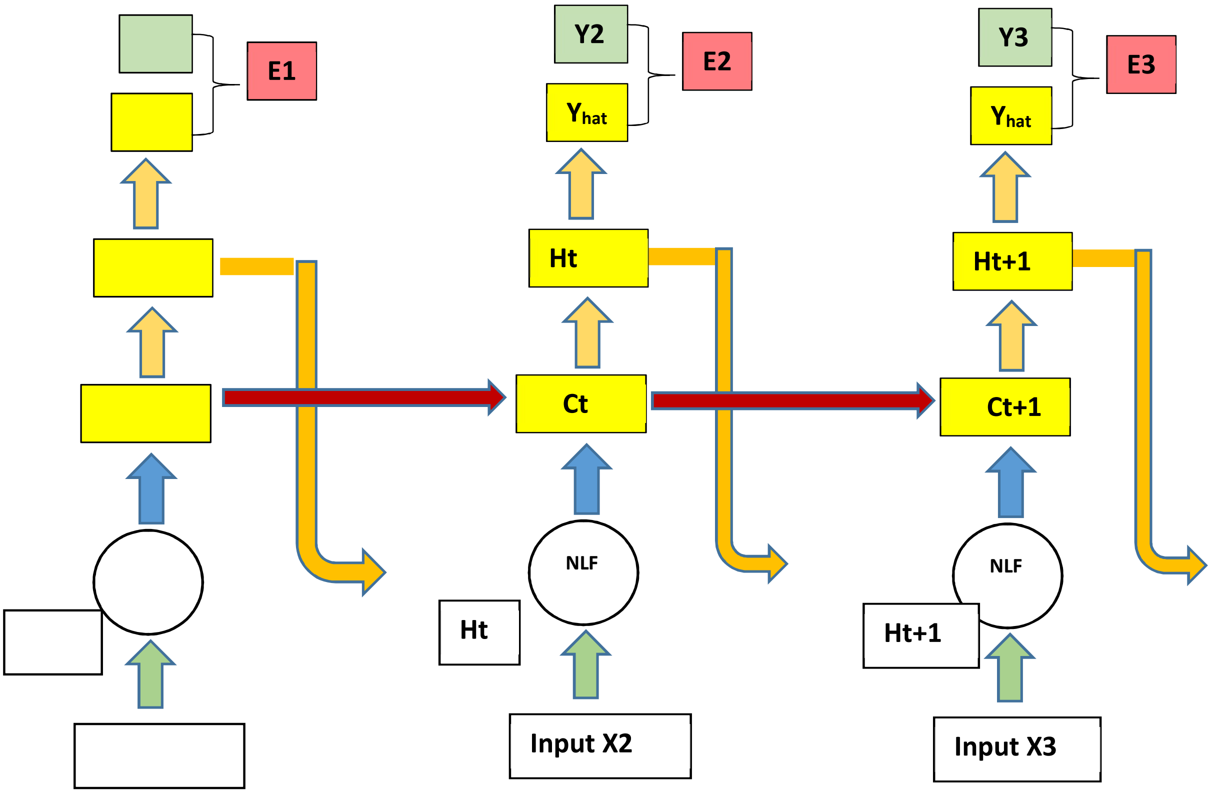
No	Equation	Gate/ Purpose
1	$F = \sigma(X_t \times w_f + H_t \times v_f + b_f)$	Forget Gate, Given a word, relevant context
2	$I = \sigma(X_t \times w_i + H_t \times v_i + b_i)$	Input Gate, given Word, probability of context
3	$G = tanh(X_t \times w_g + H_t \times v_g + b_g)$	Given word, context correlation
4	$C_{t+1} = F * C_t + I * G$	Update Cell State with mutual information
5	$O = \sigma(X_t \times w_o + H_t \times v_o + b_o)$	Context probability given the word
6	$H_{t+1} = O * tanh(C_{t+1})$	Update Hidden state with Tanh of updated Cell State and Sigmoid

Pic 24

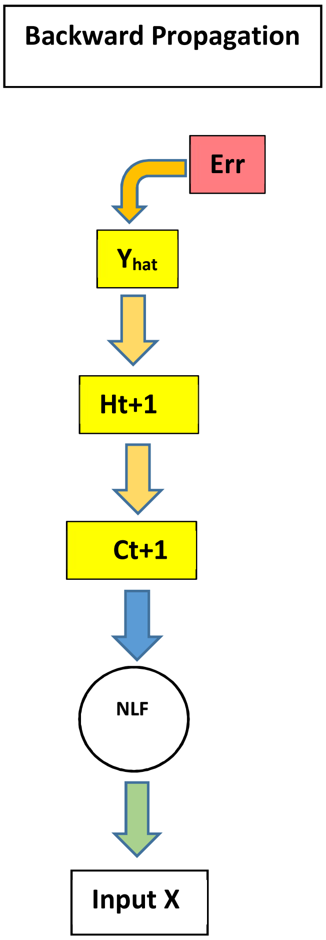
To understand the back-prop error flow, let us look at one instance of the LSTM cell first (shown below).



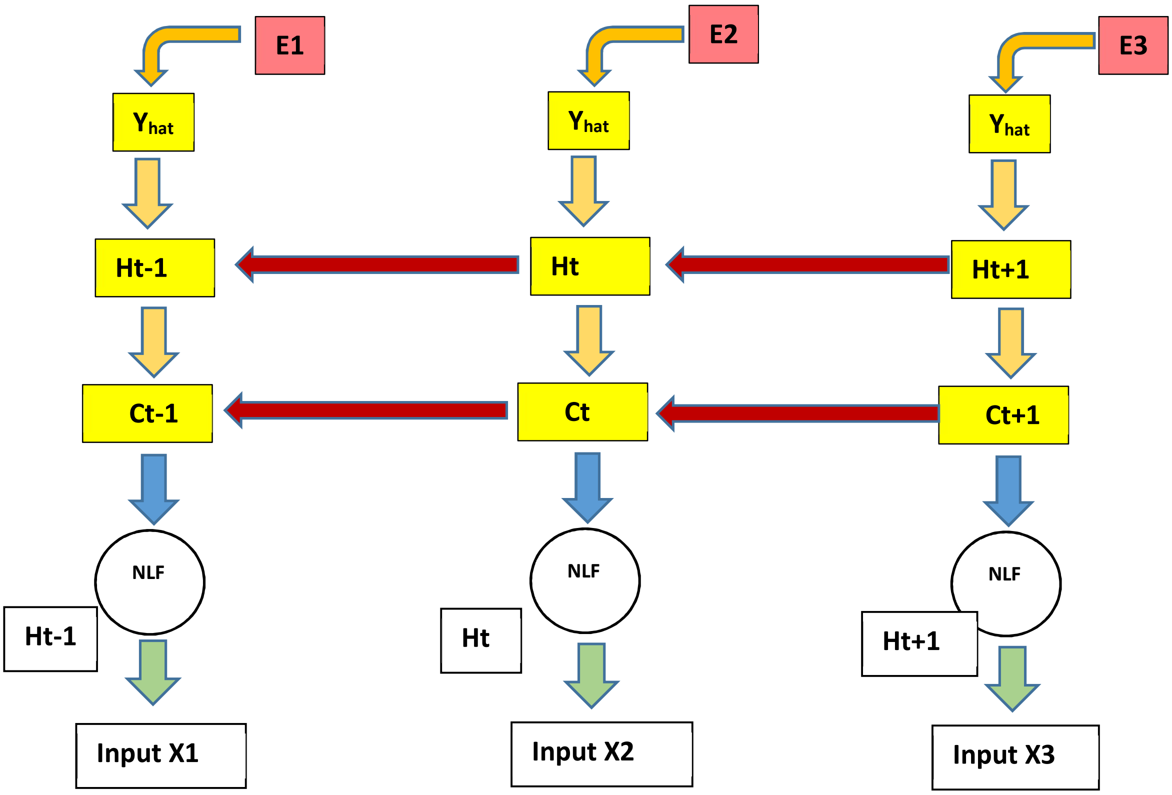
When we look at an unrolled LSTM it looks like –



LSTM Cell backward pass -



Unrolled LSTM backward pass



BackProp Error Gradient

In the unrolled LSTM the error gradients go vertically down as in RNN and horizontally like in RNN from a given time step to previous time step. But this time, we have two paths for the error gradient to go back in time. One for the C vector and one for the H vector.

1. Let the weights associated with the various gates be W (Wf , Wi , Wc , Wo).
2. Let us consider the final instance error to begin with. Let the loss function be L
3. dL/dw = d(L))/d(h_t) * d(h_t) / d(c_t) * d(c_t) /d(c_{t-1})……d(c_1)/d(w)
4. = d(L))/d(h_t) * d(h_t) / d(c_t) * (for k = 2 to last instance d(c_k) /d(c_{k-1})) * d(c_1) /d(w) -→ Eq1
5. We also know that -
6. Ct = (Ct-1 * sigmoid(Wf * [ht-1 , Xt]) + tanh(Wc* [ht-1 , Xt] * sigmoid(Wi * [ht-1 , Xt]) →Eq2
7. d(ct) / d(ct-1) =
8. d(Ct-1 * sigmoid(Wf * [ht-1 , Xt]) + tanh(Wc* [ht-1 , Xt] * sigmoid(Wi * [ht-1 , Xt]))/ d(ct-1)
9. = sigmoid(Wf * [ht-1 , Xt] + d(tanh(Wc* [ht-1 , Xt] * sigmoid(Wi * [ht-1 , Xt]) / d(ct-1) → Eq3
10. In this operation the green part can range from 0 to 1. This can go to a small fraction but not larger than 1. What prevents the gradients from vanishing is the part in red i.e. sigmoid(Wf * [ht-1 , Xt]
11. Plugging this back into the Eq 1
12. d(L)/dw = d(L))/d(h_t) * d(h_t) / d(c_t) * (for k = 2 to last instance sigmoid(Wf * [ht-1 , Xt]) * d(c_1) /d(w)

This indicates that the **Forget Gate** plays an important role in the magnitude of error gradients. During the learning phase if the weights associated with H vector and input vector are set such that the gradient remains close to 1 then the Eq 3 with red and green part will never get into vanishing gradient.

Further given the nature of product of Tanh * Sigmoid, this will never go beyond 1. Therefore exploding gradient too is avoided.Pl. refer to the following for an in-depth introduction to BPTT (Back Propagation Thru Time)

file:///C:/Users/My%20PC/Downloads/nn_2005.pdf

https://mc.ai/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients/

Finally…

Hope you survived the roller coaster and enjoyed your coffee with brownie☺. In this post I shared with you my way of understanding this challenging concept of LSTM. I belong to the group of people who cannot understand a concept without visualization and cannot move ahead without understanding what is at hand. It took me more than a year to go thru various body of knowledge generated by great minds who worked in this field which helped visualize LSTM. I have shared those links with you and hope you will find this post and the links useful.