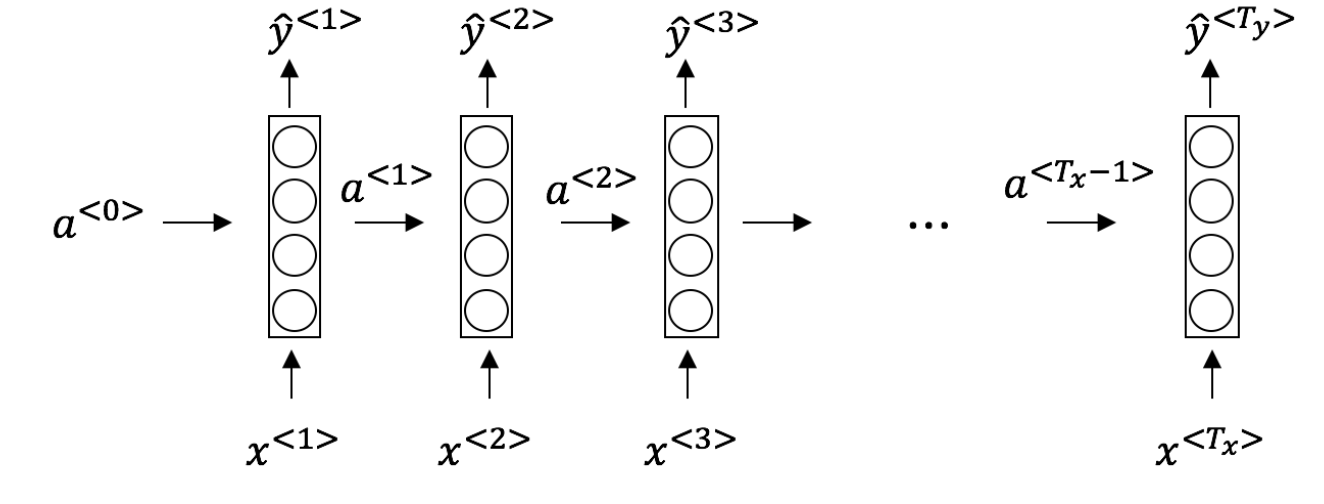
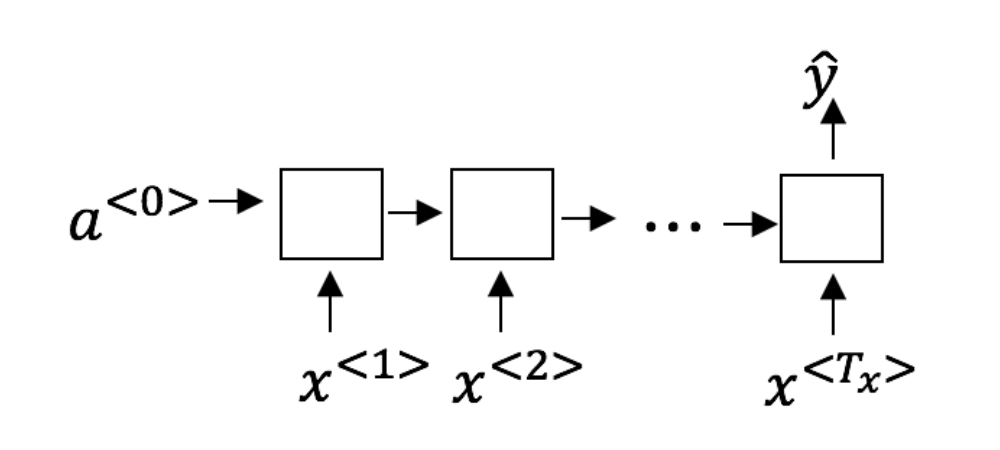
## Week 1 – RNNs

1. Suppose your training examples are sentences (sequences of words). Which of the following refers to the word in the training example?
   1. . We index into the row first to get to the training example (represented by parentheses), then the column to get to the word (represented by brackets).
2. Consider the following RNN. This specific type of architecture is appropriate when .



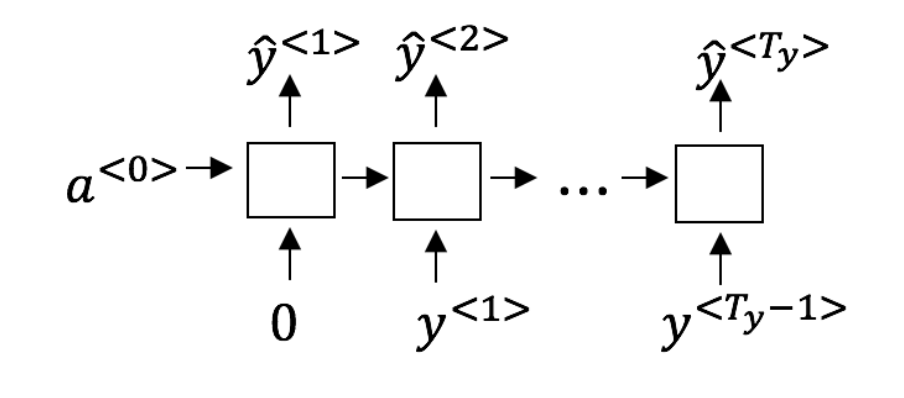
* 1. True. (It is appropriate when the input and output sequence have the same length or size.)

1. To which of the following tasks would you apply a many-to-one RNN architecture?



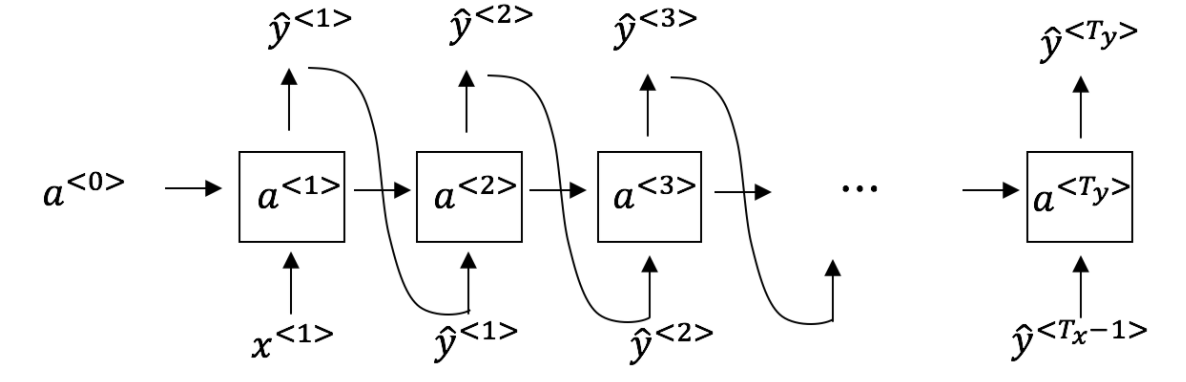
* 1. Music genre recognition.
  2. Language recognition from speech (input an audio clip and output a label indicating the language being spoken).
  3. Sentiment classification (input a piece of text and output a 0/1 to denote positive or negative sentiment.)

1. At the time step, the RNN is estimating in the following image if it is being used as the training model:



* 1. False. (In a training model, we try to predict the next steps based on the knowledge of all prior steps.)

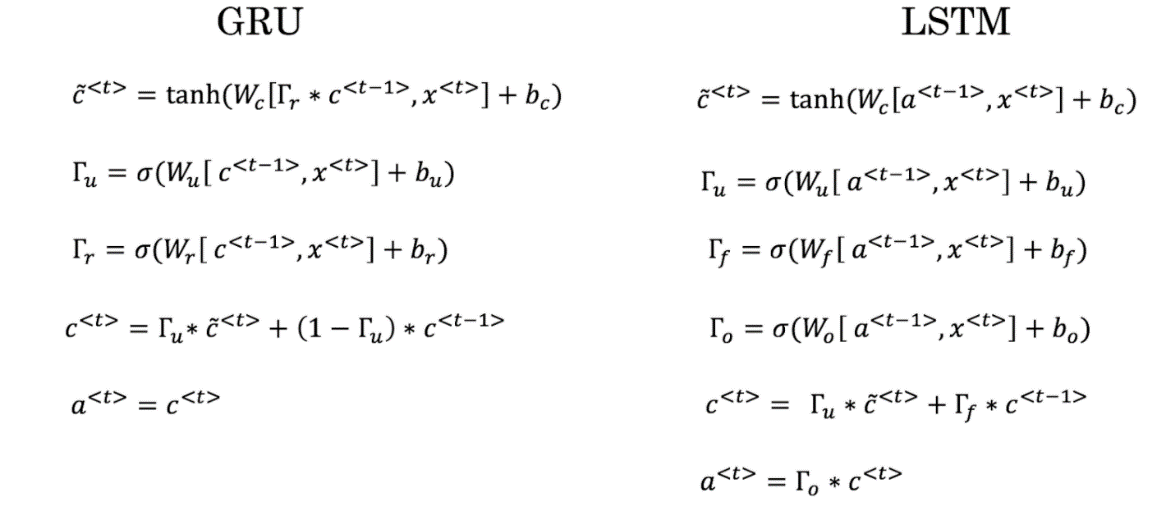
1. You have finished training a language model RNN and are using it to sample random sentences as follows:



What are you doing at each time step ?

* 1. Using the probabilities output by the RNN to randomly sample a chosen word for that time-step as .
  2. Passing the selected word to the next time-step.

1. If you are training a RNN model, and you find that your weights and activations are all taking on the value of NaN (“Not a Number”), then you have an exploding gradient problem.
   1. True. (Exploding gradients occur when large error gradients accumulate and result in very large updates to the NN model weights using training. These weights can become too large and cause an overflow, identified as NaN.)
2. Suppose you are training a LSTM. You have an 80,000-word vocabulary, and are using a LSTM with 800-dimensional activations . What is the dimension of at each time step?
   1. 800. ( is a vector of dimension equal to the number of hidden units in the LSTM.)
3. In order to simplify the Gated Recurrent Unit without vanishing gradient problems, even when training on very long sequences, you should remove the i.e. setting always.
   1. True. (If for a time step, the gradient can propagate back through that time step without much decay. For the signal to backpropagate without vanishing, we need to be highly dependent on .)
4. Using the equations for the GRU and LSTM below, the Update Gate and Forget Gate in the LSTM play a role similar to and .



* 1. False. (Instead of using to compute , a LSTM uses two gates ( and ) to compute the final value of the hidden state i.e. is used instead of .)

1. You have a pet dog whose mood is heavily dependent on the current and past few days’ weather. You’ve collected data for the past 365 days on the weather, which you represent as the sequence . You’ve also collected data on you dog’s mood, which you represent as . You’d like to build a model to map from . Should you use a unidirectional or bidirectional RNN for this problem?
   1. Unidirectional RNN, because the value of depends only on and not (i.e mood is contingent on the current and past few days’ weather, not the current, past and FUTURE days’ weather.)

## Week 2 – NLP and Word Embeddings

1. Suppose you learn a word embedding for a vocabulary of 20000 words. Then the embedding vectors could be 1000 dimensional, so as to capture the full range of variation and meaning in those words.
   1. True. (The dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors range between 50 and 1000.)
2. What is t-SNE?
   1. A non-linear dimensionality reduction technique.
3. Suppose you download a pre-trained word embedding which has been trained on a huge corpus of text. You then use this word embedding to train an RNN for a language task of recognizing if someone is happy from a short snippet of text, using a small training set.

|  |  |
| --- | --- |
| x (input text) | y (happy?) |
| Having a great time! | 1 |
| I'm sad it’s raining. | 0 |
| I’m feeling awesome! | 1 |

Even if the word “wonderful” does not appear in your small training set, what label might be reasonably expected for the input text “I feel wonderful!”?

* 1. y=1. (Word vectors empower your model with an incredible ability to generalize. The vector for “wonderful” would contain a negative/unhappy connotation which will probably make your model classify the sentence as a "1”.)

1. Which of these equations do you think should hold for a good word embedding?
2. The most computationally efficient formula for Python to get the embedding of word 1021, if is an embedding matrix, and ​ is a one-hot vector corresponding to word 1021, is .
   1. False. (It is computationally wasteful because the element-wise multiplication will be extremely inefficient.)
3. When learning word embeddings, words are automatically generated along with the surrounding words.
   1. False. (We pick a given word and try to predict its surrounding words or vice versa.)
4. In the word2vec algorithm, you estimate , where is the target word and is a context word. and are chosen from the training set to be nearby words.
   1. True.
5. Suppose you have a 10000 word vocabulary, and are learning 100-dimensional word embeddings. The word2vec model uses the following softmax function:

After training, we should expect to be very close to ​ when and are the same word.

* 1. False.

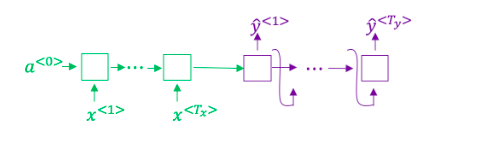
1. Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings. The GloVe model minimizes this objective:

Which of these statements are correct?

* 1. is the number of times word appears in the context of word .
  2. and should be initialised randomly at the beginning of training.
  3. Theoretically, the weighting function must satisfy .

1. You have trained word embeddings using a text dataset of ​ words. You are considering using these word embeddings for a language task, for which you have a separate labelled dataset of ​ words. Keeping in mind that using word embeddings is a form of transfer learning, under which of these circumstances would you expect the word embeddings to be helpful?

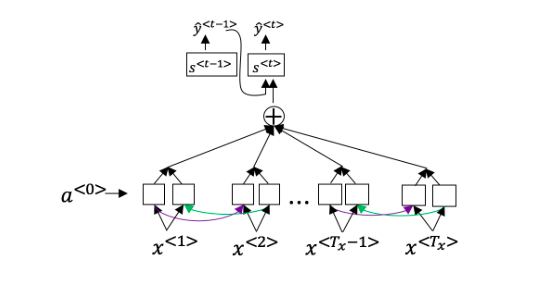
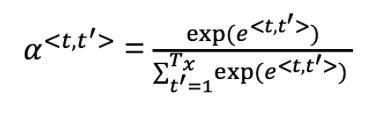
## Week 3 – Sequence Models & Attention Mechanisms

1. Consider using this encoder-decoder model for machine translation:  
     
   True/False: This model is a “conditional language model” in the sense that the decoder portion (shown in green) is modelling the probability of the input sentence .
   1. False. (The encoder-decoder model for machine translation models the probability of the output sentence conditioned on the input sentence . The encoder portion is shown in green, while the decoder portion is shown in purple.)
2. In beam search, if you increase the beam width , what would you expect to be true?
   1. Beam search will use up more memory.
   2. Beam search will run more slowly.
   3. Beam search will generally find better solutions (i.e. do a better job at maximising .
3. In machine translation, if we carry out beam search using sentence normalisation, the algorithm will tend to output overly short translations.
   1. False.
4. You are building a speech recognition system, which uses an RNN model to map from audio clip to a text transcript . Your algorithm uses beam search to try to find the value of that maximizes .  
   On a dev set example, given an input audio clip, your algorithm outputs the transcript  
    “I’m building an A Eye system in Silly con Valley.”, whereas a human gives a much superior transcript “I’m building an AI system in Silicon Valley.”  
   According to your model,

Would you expect increasing the beam width to help correct this example?

* 1. No, because indicates the error should be attributed to the RNN rather than the to the search algorithm.

1. You work on your algorithm for a few more weeks, and now find that for the vast majority of examples on which your algorithm makes a mistake,. This suggests you should focus your attention on improving the search algorithm.
   1. True.
2. Consider the attention model for machine translation.

  
And here is the formula for .  


Which of the following statements about are true?

* 1. We expect to be generally larger for values of that are highly relevant to the value the network should output for .
  2. .

1. The network learns to “pay attention” by learning the values of , which are computed using a small neural network. We can replace with as an input to this neural network because is independent of and .
   1. False. (Because depends on , which in turn depends on . So at the time we need to evaluate this network, we haven’t computed .)
2. The attention model performs the same as the encoder-decoder model, no matter the sentence length.
   1. False. (The performance of the encoder-decoder model declines as the amount of words increases. The attention model has the greatest advantage when the input sequence length is large.)
3. Under the CTC model, identical repeated characters not separated by the “blank” character (\_) are collapsed. Under the CTC model, what does the following string collapse to?

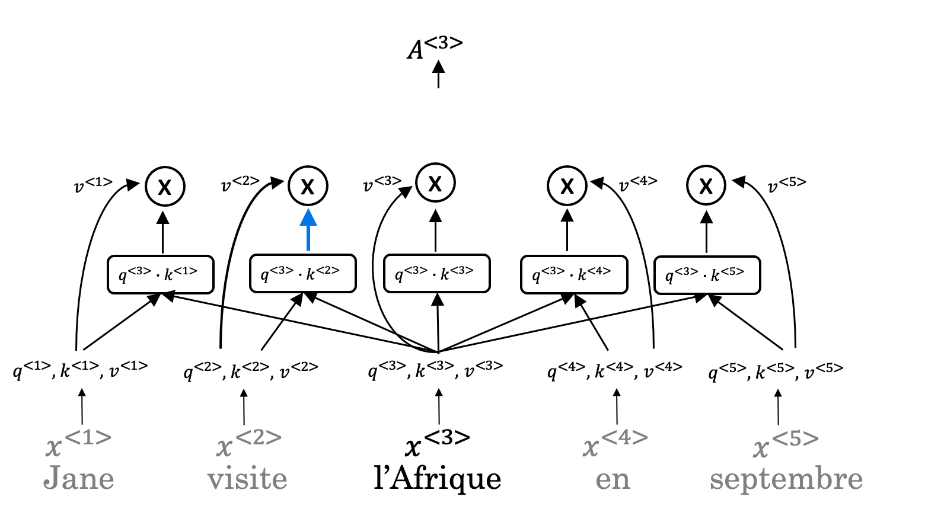
aaa\_aaaaaa\_\_\_\_\_\_\_\_rr\_ddddddddd\_\_\_\_\_\_\_v\_aaaaaa\_rrrr\_\_\_\_\_\_\_\_kk

* 1. aardvark (The basic rule for the CTC cost function is to collapse repeated characters not separated by "blank". If a character is repeated, but separated by a “blank”, it is included in the string.)

1. In trigger word detection, is:
   1. Features of the audio (such as spectrogram features) at time .

## Week 4 – Transformers

1. A Transformer Network, like its predecessors RNNs, GRUs and LSTMs, can process information one word at a time. (Sequential architecture).
   1. False. (A Transformer Network can ingest entire sentences all at the same time.)
2. Transformer Network methodology is taken from:
   1. Attention Mechanism and CNN style of processing. (Transformer architecture combines the use of attention-based representations and a CNN convolutional neural network style of processing.)
3. What are the key inputs to computing the attention value for each word?



* 1. The query, key and value.

1. What letter does the "??" represent in the following representation of Attention?
   1. K
2. Are the following statements true regarding Query (Q), Key (K) and Value (V)?

Q = interesting questions about the words in a sentence

K = specific representations of words given a Q

V = qualities of words given a Q

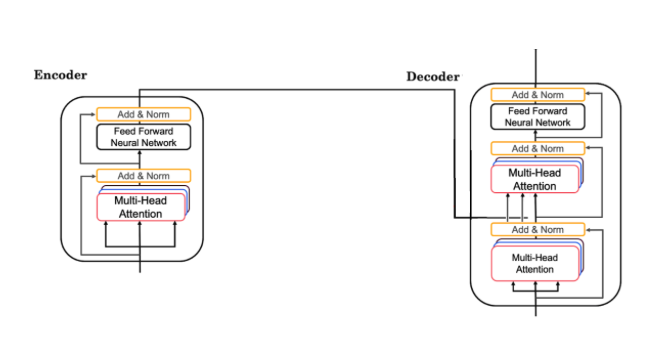
* 1. False. (Q = interesting questions about the words in a sentence, K = qualities of words given a Q, V = specific representations of words given a Q.)

1. 

here represents the computed attention weight matrix associated with the th “word” in a sentence.

* 1. False. ( here represents the computed attention weight matrix associated with the th “head” (sequence).)

1. Following is the architecture within a Transformer Network (without displaying positional encoding and output layers(s)).



What is generated from the output of the Decoder’s first block of Multi-Head Attention?

* 1. Q. (This first block’s output is used to generate the Q matrix for the next Multi-Head Attention block.)

1. The output of the decoder block contains a softmax layer followed by a linear layer to predict the next word one word at a time.
   1. False.
2. What is true?
   1. The transformer network differs from the attention model in that only the transformer network contains positional encoding. (Positional encoding allows the transformer network to offer an additional benefit over the attention model.)
3. What is **not** a good criterion for a good positional encoding algorithm?
   1. It should output a common encoding for each time-step (word’s position in a sentence).