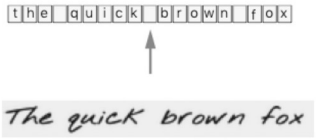
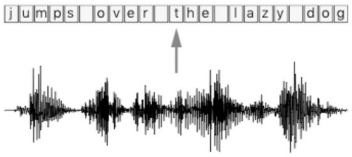
Sequence-to-Sequence Problems without known Alignment



Handwriting recognition: The input can be (x, y) coordinates of a pen stroke or pixels in an image.



Speech recognition: The input can be a spectrogram or some other frequency based feature extractor.

- fixed character rate – too restrictive
- hand alignment – very expensive
- alternative: special loss function

Supervised Learning Objective

- map from input sequence X to output sequence Y
  - X and Y can vary in length.
  - ratio of the lengths of X and Y can vary
  - no accurate alignment (correspondence of the elements) of X and Y available
- reformulated: for a given X, return an output distribution over all possible Ys

CTC: Connectionist Temporal Classification

- alignment-free
- sums over probability of all possible alignments of X and Y
- approximate solution for inference (less expensive)

CTC: Introducing ϵ (Blank) Token

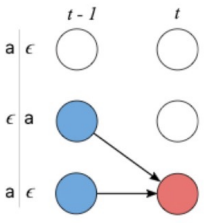
- does not correspond to anything and is simply removed from the output

CTC Alignments Properties

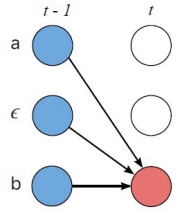
- **monotonic** (left-to-right model) – no re-ordering (unlike neural machine translation)
- **many-to-one** – many inputs can map to the same output (however many outputs cannot map to a single input)
- **len(Y) ≤ len(X)**

Recall: CTC does not find a single alignment, sums over all possible alignments

Allowed Transitions



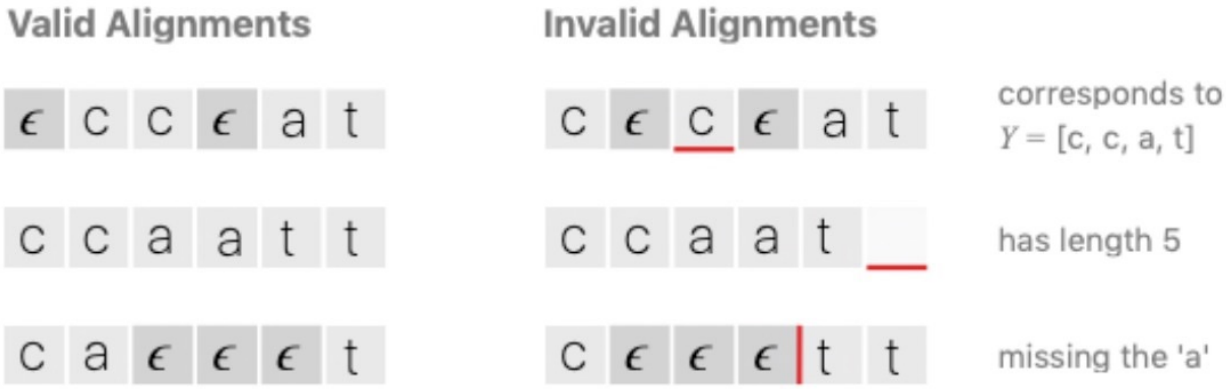
- no skip transition allowed for
  - non-blank symbols
  - blanks between same symbol



- skip transition allowed
  - previous token is blank between different symbols

Example

- output Y = [c,a,t]
  input length len(X) = 6



CTC Loss Function

We start with an input sequence, like a spectrogram of audio.

The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ} for each input step.

With the per time-step output distribution, we compute the probability of different sequences

By marginalizing over alignments, we get a distribution over outputs.

CTC Loss (work together!)

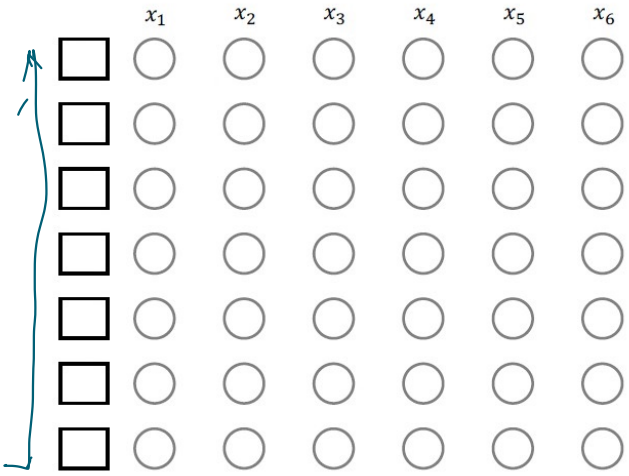
- List all valid alignments of length 6 for the target sequence “door”!
  - basis: doϵor (double o requires ϵ)
  - only possible to (a) add an ϵ somewhere or (b) double a normal char

a) ϵdoϵor dϵoϵor doϵϵor doϵoϵr doϵorϵ

b) ddoϵor dooϵor doϵoor doϵorr

CTC Loss (work together!)

Consider a sequence aab which is used to train a model with CTC loss. Draw all connections in the following CTC graph structure which correspond to valid sequences for the 6 time steps below. Use the empty rectangles on the left to write the respective character.



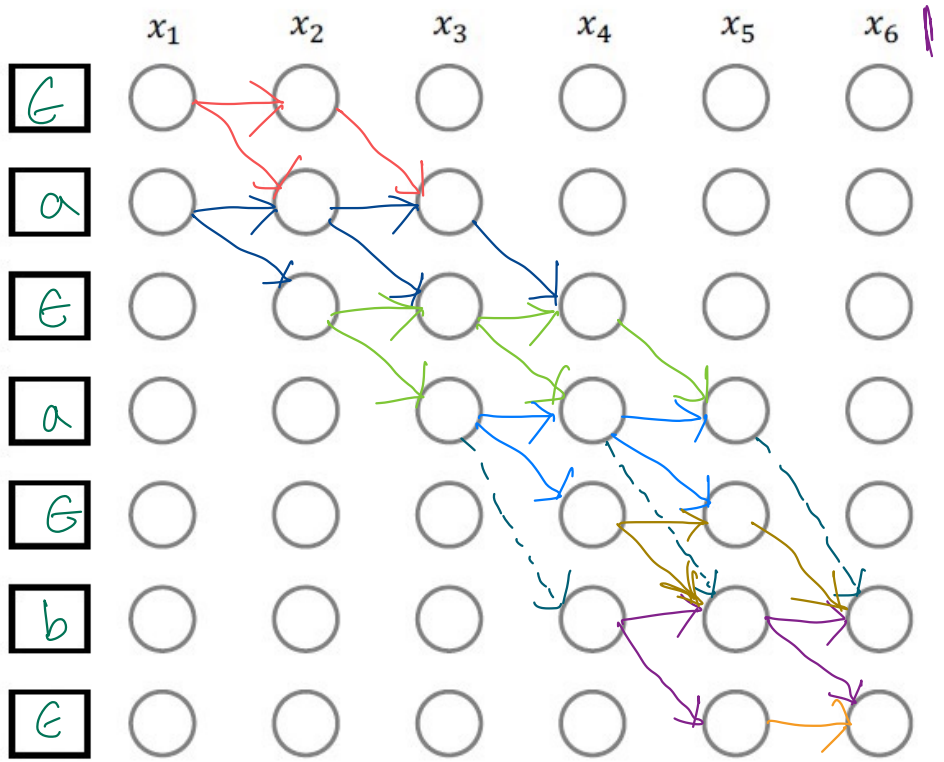
aab

ϵaaϵbϵ

ϵaϵabϵϵ

CTC Loss (work together!)

Consider a sequence aab which is used to train a model with CTC loss. Draw all connections in the following CTC graph structure which correspond to valid sequences for the 6 time steps below. Use the empty rectangles on the left to write the respective character.

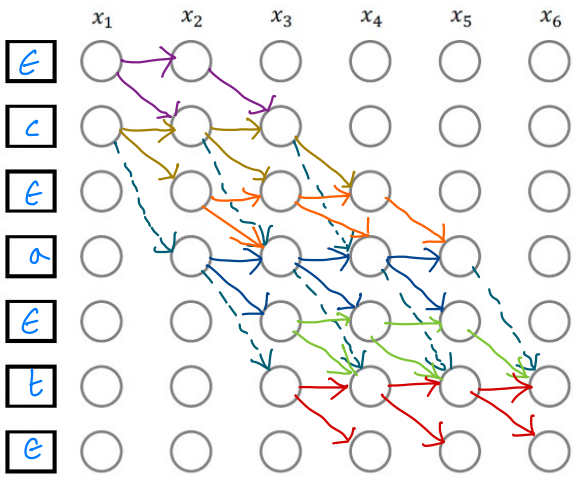


Given aab
ind_seq_Len=7
(CTC)
ϵaϵaϵbϵϵ

skip connection

CTC Loss (work together!)

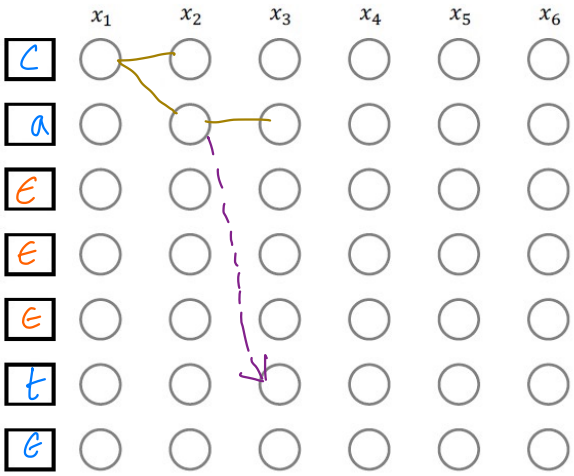
Consider a sequence aab which is used to train a model with CTC loss. Draw all connections in the following CTC graph structure which correspond to valid sequences for the 6 time steps below. Use the empty rectangles on the left to write the respective character.



cat

CTC Loss (work together!)

Consider a sequence aab which is used to train a model with CTC loss. Draw all connections in the following CTC graph structure which correspond to valid sequences for the 6 time steps below. Use the empty rectangles on the left to write the respective character.



CTC Loss (work together!)

Consider a sequence aab which is used to train a model with CTC loss. Draw all connections in the following CTC graph structure which correspond to valid sequences for the 6 time steps below. Use the empty rectangles on the left to write the respective character.