# Towards End-to-End Speech Recognition with Recurrent Neural Networks

Alex Graves, Navdeep Jaitly

# Introduction

In previous speech recognition, the neural networks are at present only a single component in a complex pipeline

Shortage of previous speech recognition system:

The frame-level training targets must be inferred from the alignments determined by the HMM

A pronunciation dictionary is necessary to map from words to phoneme sequences

Create pronunciation dictionary costs a lot of labor

The quality of the dictionary affects the result of speech recognition system dramatically

Alignment

Pronunciation dictionary

Improvement of this paper

Using a deep bidirectional LSTM network with a Connectionist Temporal Classification output layer

No pronunciation dictionary is necessary

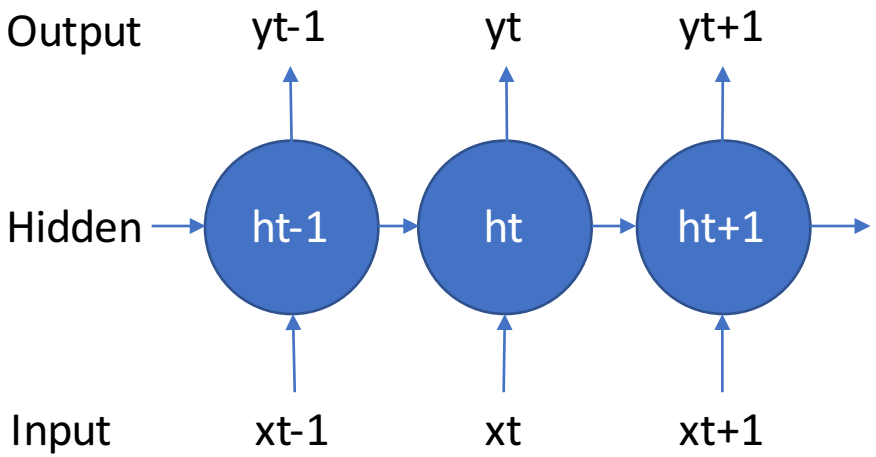
CTC integrates out over all possible input-output alignments, no forced alignment is required to provide training targets

# Bidirectional Recurrent Neural Networks

In the speech recognition, we need consider the information from both past and future

Add backward Layer to Standard RNN to construct Bidirectional RNN

Standard RNN

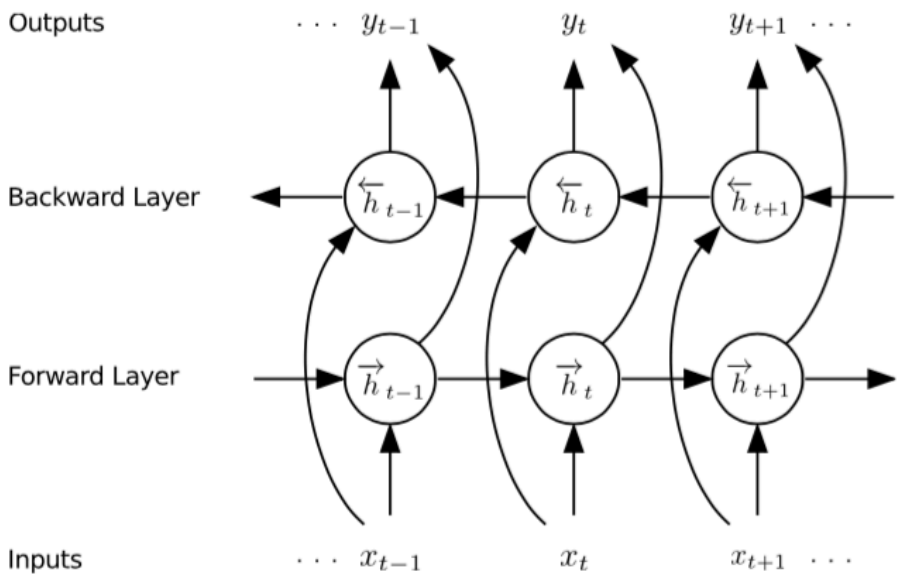



$$h_t = \mathcal{H}\left(W_{ih}x_t + W_{hh}h_{t-1} + b_h\right) \quad (1)$$

$$y_t = W_{ho}h_t + b_o \quad (2)$$

Input vector x=(x1,…,xT)
Hidden vector h=(h1,…,hT)
Output vector y=(y1,…,yT)
Optimize Wih Whh and Who

Bidirectional RNN




$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \quad (8)$$

$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right) \quad (9)$$

$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_o \quad (10)$$

Graves Towards End-to-End Speech Recognition with Recurrent Neural Networks

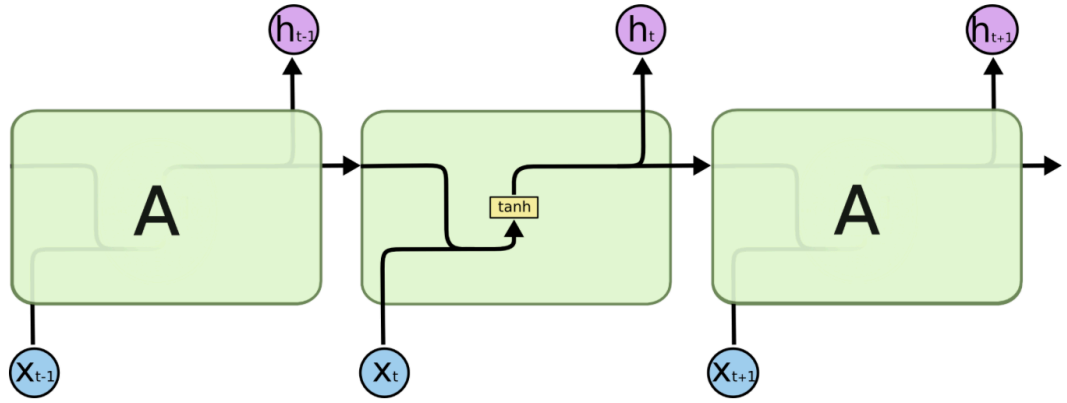
http://x-algo.cn/index.php/2016/04/25/rnn-recurrent-neural-networks-derivation-and-implementation/

http://www.shareditor.com/blogshow/?blogId=116

# Long Short Term Memory

Standards RNN performs bad for long-term dependencies
because gradients prorogation over many stages tend to either vanish or explode
In practice, Long Short Term Memory is widely used to preserve long-term dependencies

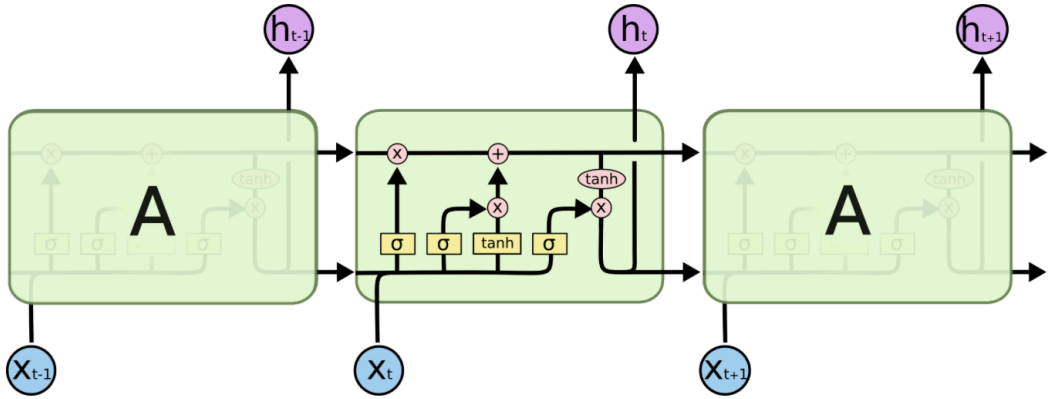
Standard RNN



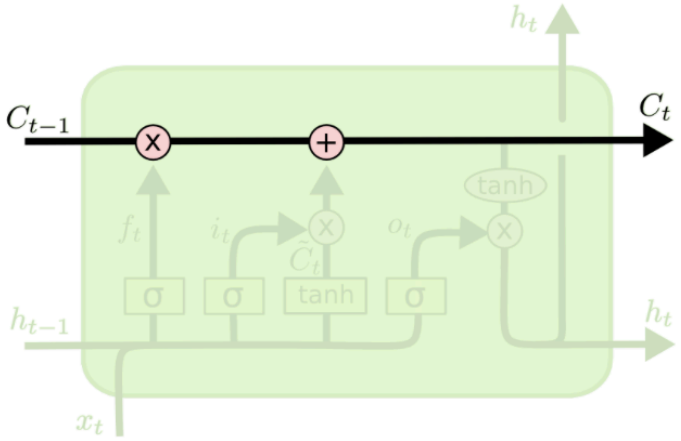

The repeating module in a standard RNN contains a single layer.

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.
It's very easy for information to just flow along it unchanged.

LSTM



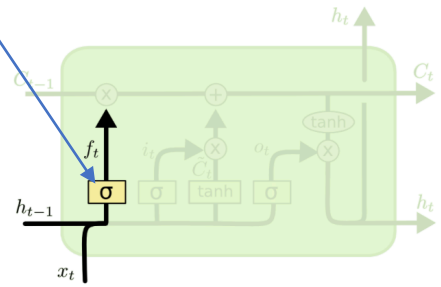

The repeating module in an LSTM contains four interacting layers.




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

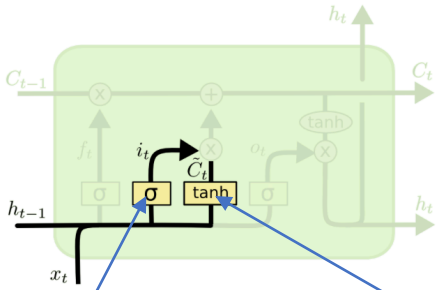
# Long Short Term Memory

Forget Gate




$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

The first step in our LSTM is to decide what information we're going to throw away from the cell state.




$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

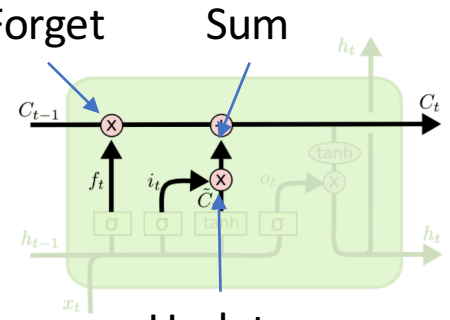
The next step is to decide what new information we're going to store in the cell state.

Input gate : which values we'll update

New candidate values, added to the state

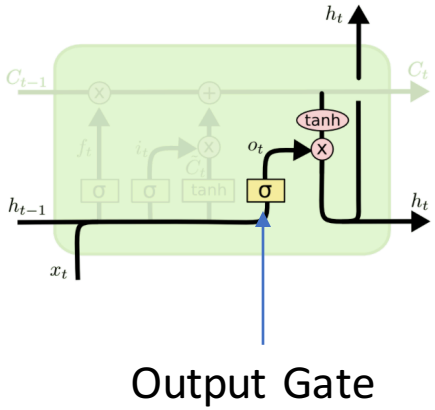
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

# Long Short Term Memory




$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Update the old cell state Ct-1, into the new cell state Ct



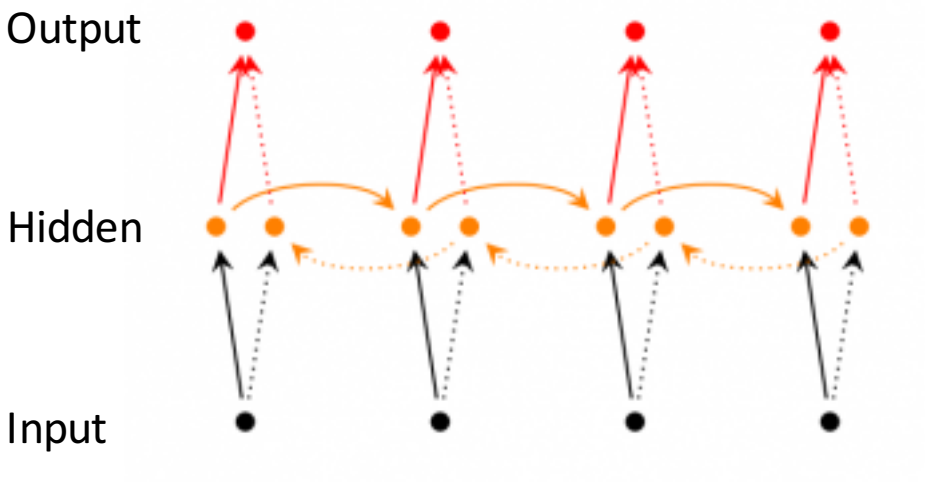

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Finally, we need to decide what we're going to output. This output will be based on our cell state, but will be a filtered version.

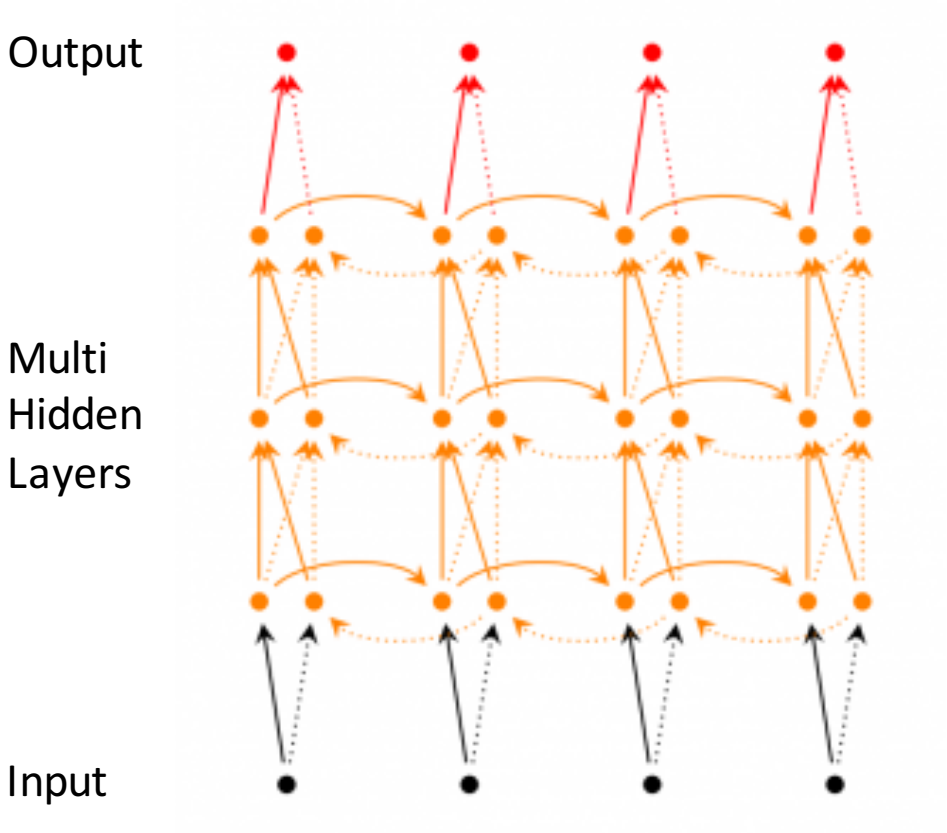
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

# Deep Recurrent Neural Network

Bidirectional RNN




Deep Bidirectional RNN




More ability to learning

http://blog.csdn.net/heyongluoyao8/article/details/48636251

# Connectionist Temporal Classification (CTC)

**Typical frame-level classifier:**

Requires a training target for every frame in the audio

(1)circular dependency:

good alignment <--> good classifier

(2) no need for character-level alignment

**Using CTC as objective function:**

Allows an RNN to be trained without any prior alignment between the input and the transcription

# Connectionist Temporal Classification (CTC)

**x** : Input sequence with length T
**a** : output sequence of blank and label indices (CTC alignment)
**y** : output transcription
**B** : matching CTC alignment to transcription

Example:
$a_1$ = (a, -, b, c, -, -)  $a_2$ = (-, -, a, -, b, c)
$a_3$ = (a, -, b, -, c, c)  $a_4$ = (a, a, b, b, c, c)

y = (a, b, c)

$B(a_1) = B(a_2) = B(a_3) = B(a_4) = y$

$$\Pr(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{a}\in\mathcal{B}^{-1}(\boldsymbol{y})} \Pr(\boldsymbol{a}|\boldsymbol{x})$$

$$CTC(\boldsymbol{x}) = -\log \Pr(\boldsymbol{y}^*|\boldsymbol{x})$$

# Expected Transcription Loss

**L(x, y) :** transcription loss function (e.g. WER)
**L(x) :** expected transcription loss

$$\mathcal{L}(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \Pr(\boldsymbol{y}|\boldsymbol{x})\mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$$

Approximate with Monte-Carlo sampling

$$\mathcal{L}(\boldsymbol{x}) \approx \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(\boldsymbol{x}, \mathcal{B}(\boldsymbol{a}^i)), \quad \boldsymbol{a}^i \sim \Pr(\boldsymbol{a}|\boldsymbol{x})$$

Differentiate L(x) with respect to RNN outputs

$$\frac{\partial \mathcal{L}(\boldsymbol{x})}{\partial \Pr(k, t|\boldsymbol{x})} \approx \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(\boldsymbol{x}, \mathcal{B}(\boldsymbol{a}^{i,t,k}))$$

# Expected Transcription Loss

$$\frac{\partial \mathcal{L}(\boldsymbol{x})}{\partial y_t^k} \approx \frac{\Pr(k,t|\boldsymbol{x})}{N} \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{x}, \mathcal{B}(\boldsymbol{a}^i)) - \mathcal{Z}(\boldsymbol{a}^i, t)$$

where

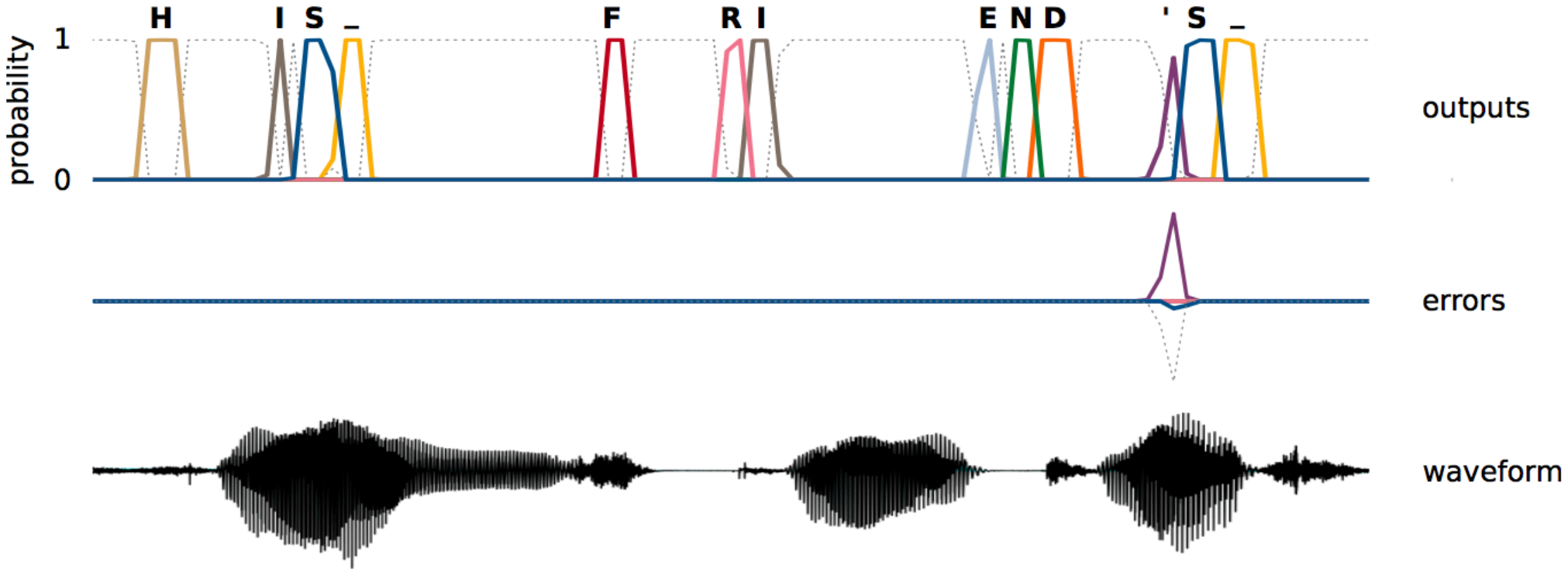
$$\mathcal{Z}(\boldsymbol{a}^i, t) = \sum \Pr(k', t|\boldsymbol{x}) \mathcal{L}(\boldsymbol{x}, \mathcal{B}(\boldsymbol{a}^{i,t,k'}))$$

The derivative added to $y^k_t$ equals to
the difference between
the loss with $a^i_t = k$
and the expected loss with $a^i_t$ sampled from Pr(k', t|x)

Encourage outputs changing that alter the loss functions

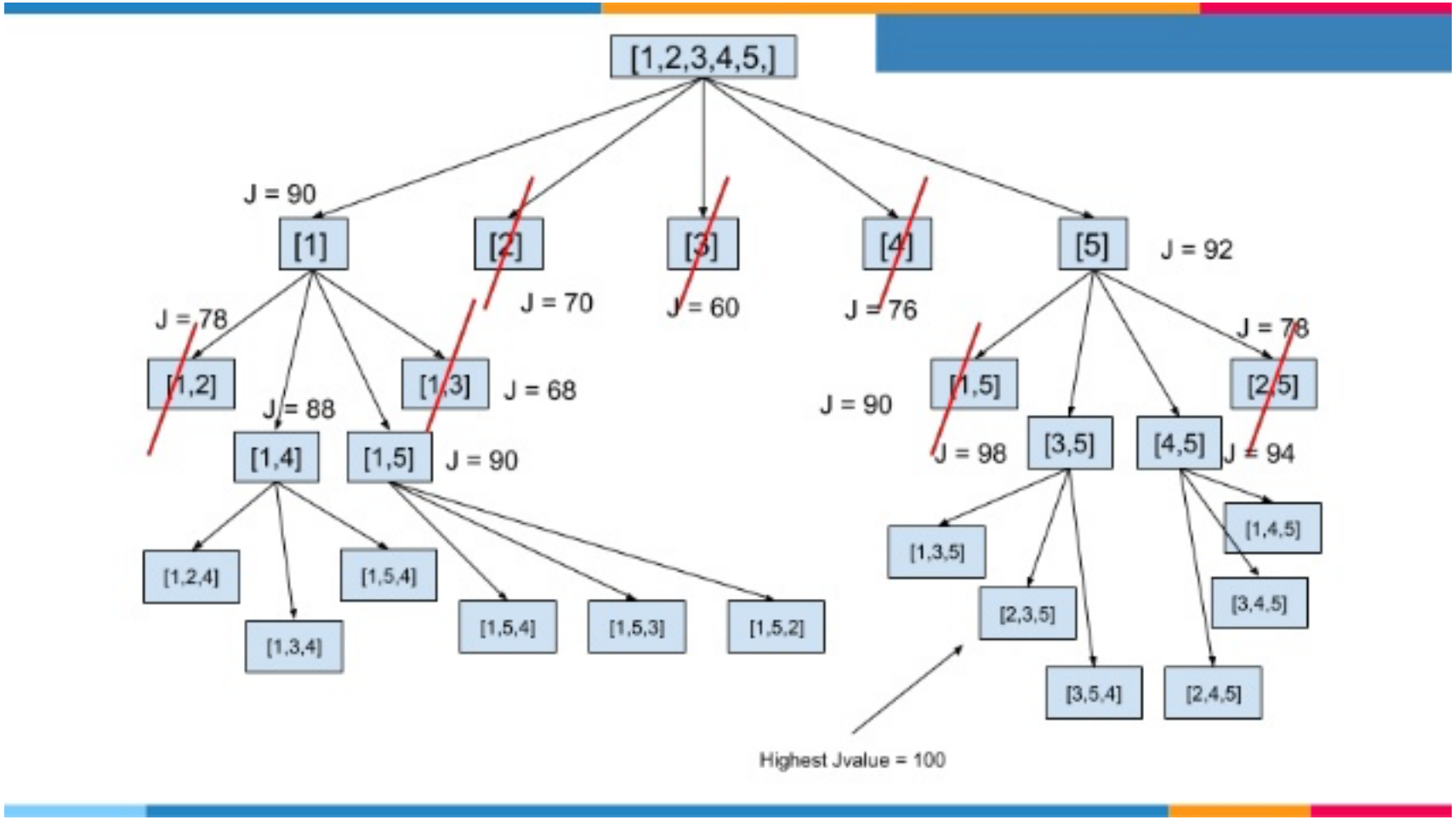
# Expected Transcription Loss

Target: HIS_FRIENDS_
Output: HIS_FRIEND'S_
Changing of the apostrophe is encouraged



# Decoding

Decode the output of CTC to a sequence of result: beam search algorithms




Doing pruning doing the search and cut the nodes with less probability and only keep top n nodes

http://www.slideshare.net/ChaminNalindaLokuGam/branch-and-bound-and-beam-search-feature-selection-algorithms

# Experiments

*Table 1.* **Wall Street Journal Results.** All scores are word error rate/character error rate (where known) on the evaluation set. 'LM' is the Language model used for decoding. '14 Hr' and '81 Hr' refer to the amount of data used for training.

| SYSTEM | LM | 14 HR | 81 HR |
|---|---|---|---|
| RNN-CTC | NONE | 74.2/30.9 | 30.1/9.2 |
| RNN-CTC | DICTIONARY | 69.2/30.0 | 24.0/8.0 |
| RNN-CTC | MONOGRAM | 25.8 | 15.8 |
| RNN-CTC | BIGRAM | 15.5 | 10.4 |
| RNN-CTC | TRIGRAM | 13.5 | 8.7 |
| RNN-WER | NONE | 74.5/31.3 | 27.3/8.4 |
| RNN-WER | DICTIONARY | 69.7/31.0 | 21.9/7.3 |
| RNN-WER | MONOGRAM | 26.0 | 15.2 |
| RNN-WER | BIGRAM | 15.3 | 9.8 |
| RNN-WER | TRIGRAM | 13.5 | 8.2 |
| BASELINE | NONE | — | — |
| BASELINE | DICTIONARY | 56.1 | 51.1 |
| BASELINE | MONOGRAM | 23.4 | 19.9 |
| BASELINE | BIGRAM | 11.6 | 9.4 |
| BASELINE | TRIGRAM | 9.4 | 7.8 |
| COMBINATION | TRIGRAM | — | 6.7 |

Performance is close to the baseline, while baseline has a lot of priority knowledge.
Improve 1% when combine the model

Combine RNN with DNN

# Discussion

Examples: (NO dictionary or language model used for decoding)

target: **TO ILLUSTRATE** THE POINT A PROMINENT MIDDLE EAST ANALYST IN WASHINGTON RECOUNTS A CALL FROM ONE CAMPAIGN
output: **TWO ALSTRAIT** THE POINT A PROMINENT MIDILLE EAST ANALYST IM WASHINGTON RECOUNCACALL FROM ONE CAMPAIGN

target: T. W. A. ALSO PLANS TO HANG ITS **BOUTIQUE SHINGLE** IN AIRPORTS AT LAMBERT SAINT
output: T. W. A. ALSO PLANS TOHING ITS **BOOTIK SINGLE** IN AIRPORTS AT LAMBERT SAINT

target: ALL THE EQUITY RAISING IN MILAN GAVE THAT STOCK MARKET INDIGESTION LAST YEAR
output: ALL THE EQUITY RAISING IN MULONG GAVE THAT STACRK MARKET IN TO JUSTIAN LAST YEAR

target: THERE'S UNREST BUT WE'RE NOT GOING TO LOSE THEM TO DUKAKIS
output: THERE'S UNREST BUT WERE NOT GOING TO LOSE THEM TO DEKAKIS

# Thanks!