# Understanding Recurrent Neural Networks

May 19, 2017

**Abstract**

Recurrent Neural Networks have been used with tremendous success for various problems ranging from Video Captioning, natural language modeling, translation etc. There have also been attempts for using Recurrent Neural Network structure for lossless compression [Gra13]. Variants of the Vanilla RNN, such as LSTM, GRU networks, give a deeper dependency profile, which is useful for modeling more complex distributions.

We try to obtain some performance bounds for the RNN. We also argue that for some sources (which are not too uncommon), the recurrent neural network framework might not lead to the optimal performance.

## 1 RNN Framework

For the purpose of analysis, we restrict ourselves to the problem of language modeling using a Character level model. The generic Character level-RNN model can be viewed as follows:

1. The input is a sequence of values $x_1, x_2, \ldots, x_N, \ldots$ generated from a stationary distribution, with entropy rate of $\mathbb{H}(X)$.
2. The Char-RNN model takes in an input as $x_i$ and the State $s_i$ at that point, and generates a probability distribution estimate for the next symbol $\hat{P}(X_{i+1})$.
3. The Performance is measured using the standard Log-Loss (Softmax loss, in deep learning literature). $L_i = \log \frac{1}{\hat{P}(X_{i+1}=x_{i+1})}$. The overall normalized: $L = \frac{\sum_{i=1}^{N} L_i}{N}$

We do not restrict the computations inside the RNN-Block. Our analysis is based only on the framework of the RNN model.

### 1.1 Compression perspective

The Character-level model with the log-loss can be viewed with a different perspective of compression, as shown in Fig[1]. It is seen that the optimal log-loss achievable is the entropy rate. Thus, we can view this problem as compression problem, with the following circuit shown in Fig[1].

$$\begin{aligned}
\lim_{N\to\infty} L &= \lim_{N\to\infty} \frac{\sum_{i=1}^{N} L_i}{N} \\
&= \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \log \frac{1}{\hat{P}(X_{i+1} = x_{i+1})} \\
&= \lim_{N\to\infty} \frac{1}{N} \log \frac{1}{\hat{P}(x^N)} \\
&= \mathbb{H}(X) + D(P||\hat{P}) \\
&\geq \mathbb{H}(X)
\end{aligned}$$

To achieve the entropy rate $\mathbb{H}(X)$ log-loss (or equivalently compression), we need to predict the distribution $\hat{P}(X_{i+1})$ correctly at every timestep, i.e for every $i \in [1, N]$

$$\hat{P}(X_{i+1}|x_i, s_i) = P(X_{i+1}|X_{-\infty}^{i}) \tag{1}$$

In the next section, we analyze the average State size required to achieve this performance.

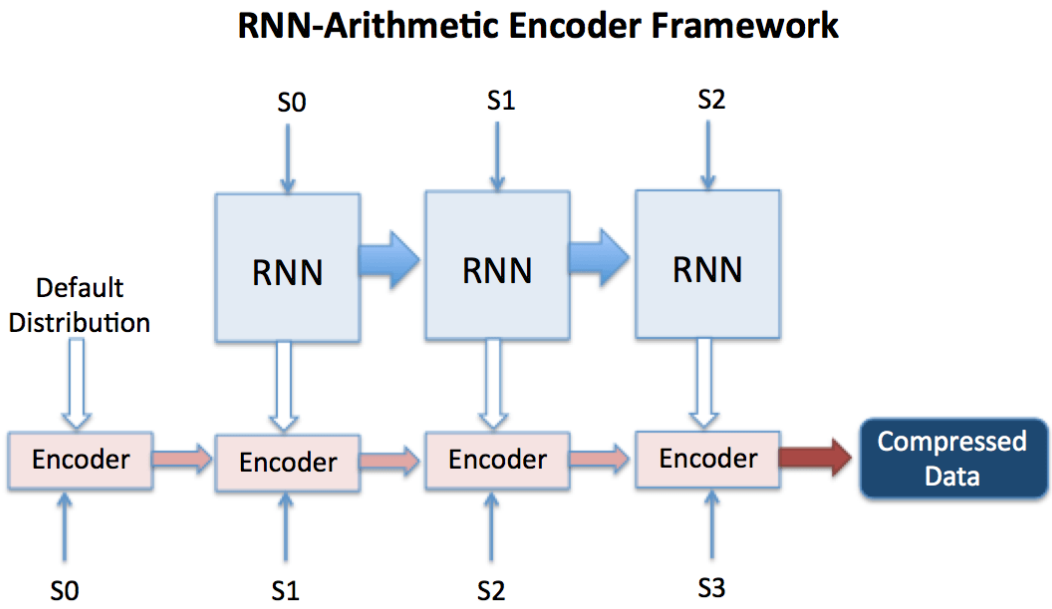


Figure 1: An equivalent view of the Char-RNN model

## 2 Performance Analysis

We claim the following:

**Lemma 1.** *The minimum state size required to achieve optimal performance is quantified by the excess information,* $I(X_{-\infty}^{-1}; X_1^{\infty}|X_0)$

*Proof.* (more of a justification) The intuition for the formula is as follows: to predict correctly for $X_1^{\infty}$, we need to have the information about the same from past $X_{-\infty}^{-1}$ (for example, dependency etc.). All this has to pass through the state $S_0$.

Note that the block which has input $X_0$ will only use a fraction of the information in the state $S_0$, and passes on the rest of it ahead (by modifying some parts of it).

The theoretical justification is as follows:

1. Let $S_{0[i]}$ be the part of the information useful for prediction of $X_i$. For correctly predicting $\hat{P}(X_1|x_0, s_{0[1]}) = P(X_1|x_{-\infty}^{i})$, we do not need to describe the entire past, $x_{-\infty}^{i}$), but only the mode of the distribution which is captured by $s_{0[1]}$. Now this is essentially equivalent to the markov chain:
$$X_{-\infty}^{-1} \rightarrow S_{0[1]} \rightarrow X_1(given X_0)$$
Thus, given $X_0, S_{0[1]}$ should make $X_1$, independent of the past. To communicate $S_{0[1]}$, we require $H(S_{0[1]}|X_0)$ bits, which can be bounded as:
$$\begin{aligned} H(S_{0[1]}|X_0) &= I(S_{0[1]}; X_{-\infty}^{-1}|X_0) \\ &\geq I(X_1; X_{-\infty}^{-1}|X_0) \end{aligned}$$
We use the fact that $S_{0[1]}$ is a function of the past $X_{-\infty}^{-1}$ in the first inequality. In the second inequality, we use the data processing inequality.

2. Now, for predicting $X_i$, the state $S_{0[i]}$, needs to carry atleast the information between $X_i$, and $X_{-\infty}^{-1}$, given the values $X_0^{i-1}$. Thus, by considering a similar markov chain:
$$X_{-\infty}^{-1} \rightarrow S_{0[i]} \rightarrow X_i(given X_0^{i-1})$$

Similarly we can bound $H(S_{0[i]}|X_0^{i-1})$ as:

$$\begin{aligned} H(S_{0[i]}|X_0) &= I(S_{0[i]}; X_{-\infty}^{-1}|X_0^{i-1}) \\ &\geq I(X_i; X_{-\infty}^{-1}|X_0^{i-1}) \end{aligned}$$

3. Now, to find the overall bits required, we sum over $H(S_0[i]|X_0)$ for all $i$.

$$\begin{aligned} H(S_0|X_0) &= \sum_{i=1}^{\infty} H(S_{0[i]}|X_0) \\ &\geq \sum_{i=1}^{\infty} I(X_i; X_{-\infty}^{-1}|X_0^{i-1}) \\ &= I(X_{-\infty}^{-1}; X_1^{\infty}|X_0) \end{aligned}$$

This, justifies the theorem, as on average we require $I(X_{-\infty}^{-1}; X_1^{\infty}|X_0)$ bits of state information. If the state size is less, we cannot perfectly obtain the distributions correctly, even with any kind of improved RNN framework (such as LSTM, GRU). □

## 2.1 Understanding Excess Information

Before we go on to apply the Lemma, let us first understand the excess information quantity.

$$\begin{aligned} I(X_{-\infty}^{-1}; X_1^{\infty}|X_0) &= \lim_{N\to\infty} I(X_{-\infty}^{-1}; X_1^{N}|X_0) \\ &= \lim_{N\to\infty} H(X_1^N|X_0) - H(X_1^N|X_{-\infty}^{0}) \\ &= \lim_{N\to\infty} \sum_{i=1}^{N} \left[H(X_0|X_{-i}^{-1}) - \mathbb{H}(X)\right] \end{aligned}$$

The second equality occurs due to stationarity of $X$. Notice that essentially the excess information term captures the convergence rate to the entropy rate $\mathbb{H}(X)$ of the process, as $\lim_{N\to\infty} H(X_0|X_{-N}^{-1}) = \mathbb{H}(X)$.

This, gives us a very nice way of analyzing and computing the excess information $I(X_{-\infty}^{-1}; X_1^{\infty}|X_0)$ for common processes. We consider a few examples next.

## 2.2 Examples:

### IID Model

If $X_1, X_2, \ldots, X_N$ are independent and identically distributed samples, then as $H(X_0|X_{-N}^{-1}) = H(X_0)$, there is immediate convergence to the entropy rate. Thus:

$$I(X_{-\infty}^{-1}; X_1^{\infty}|X_0) = 0$$

Now, even in the RNN model, it is clear that with no state information at all, we should be able to do a good job of compression by only looking at the past symbol (infact, we do not need to look at any symbol).

### Markov Model

Let us consider a Markov process. For a markov process: $H(X_0|X{-}1) = \mathbb{H}(X)$. Thus, again in this case,

$$I(X_{-\infty}^{-1}; X_1^{\infty}|X_0) = 0$$

This again matches our analysis in the RNN case. As for prediction of $X_i$, we can base our decision on $X_{i-1}$, as it is available, thus, we should not require any state information.

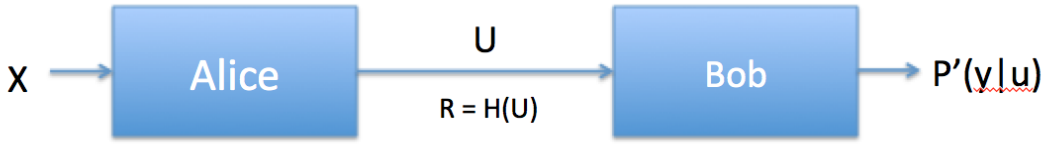


Figure 2: The Limited Side-Information based compression problem

**k-Markov Model**

We consider the 0-entropy source generated by the process:

$$X_N = X_{N-1} + X_{N-k} \tag{2}$$

As this is a deterministic process, once we fix the initial $k$ bits, this the entropy rate is $\mathbb{H}(X) = 0$. Notice that, $H(X_0|X_{-r}^{-1}$ is going to be 0 for $r \geq k$, since we can exactly predict $X_0$ then. Thus,

$$I(X_{-\infty}^{-1}; X_1^{\infty}|X_0) \leq k-1$$

This matches our intuition that $k-1$ of state information is sufficient for perfect prediction on the k-Markov sequence (since, you need to essentially send the previous k-1 bits to predict the entire sequence. Anything less is not sufficient for exact prediction)

# 3 Excess Information in Real Life

We obtained bounds on the state size based on the excess information for a language (images, text, video) $I(X_{-\infty}^{-1}; X_1^{\infty}|X_0)$. First of all, notice that the RNN-compressor is a stronger class of algorithms, as it can compress more complicated sources, such as hidden markov models with finite amount of memory, which most of the existing compressors cannot (including Lempel-Ziv) etc.

Second and foremost, what about the case where the excess information is unbounded? This will the case if the convergence is not fast enough to the entropy rate. It seems that this is hypothesized to be exactly the case in Natural languages (Hilberg Conjecture) [Dęb14][D+15]. Thus, in a sense, we do not expect RNN framework to work optimally for NLP, or images, or videos (although it may come close). It remains to understand in this case:

1. How close to the optimal can we come with state size $S$
2. What structural modification (if any) will aid in achieving the optimal

# 4 Solving a simpler problem

Let us first consider a simpler information theoretic formulation of the problem above. We will try to solve (at least obtain some bounds on the problem, which will later be helpful for in understanding the recurrent neural networks problem.

The problem is as follows (See Figure 2):

*Nature generates symbols $X, Y$ distributed as $p(x, y)$. Alice receives the $X$ samples, and needs to communicate $R$ bits to Bob, so that he gets a good estimate of the conditional distribution $p(Y|x)$, which he uses for the compression of $Y$. In such a case, we aim to analyze the tradeoff between the compression performance loss, versus the rate $R$.*

**Lemma 2.** *If Alice send a message $U$, and Bob uses the distribution $\hat{p}(y|u)$ for compression, then the compression loss $D$, is:*

$$D = I(X;Y|U) + D\left(p(y|u)||\hat{p}(y|u)\right) \tag{3}$$

*Proof.*

$$\begin{aligned}
D &= \sum_{x,y,u} p(x,y,u) \log \frac{1}{\hat{p}(y|u)} - H(Y|X) \\
&= \sum_{x,y,u} p(x,y,u) \log \frac{1}{\hat{p(y|u)}} - \sum_{x,y} p(x,y) \log \frac{1}{\hat{p}(y|x)} \\
&= \sum_{x,y,u} p(x,y,u) \log \frac{1}{\hat{p(y|u)}} - \sum_{x,y,u} p(x,y,u) \log \frac{1}{\hat{p}(y|x)} \\
&= \sum_{x,y,u} p(x,y,u) \log \frac{p(y|x)}{\hat{p}(y|u)} \\
&= \sum_{x,y,u} p(x,y,u) \log \frac{p(y|x)p(y|u)}{p(y|u)\hat{p}(y|u)} \\
&= I(X;Y|U) + D\left(p(y|u)||\hat{p}(y|u)\right)
\end{aligned}$$

This result also tells that for a given message $U$, Bob is always better using the distribution $\hat{p}(y|u) = p(y|u)$. □

**Theorem 1.** *If Alice sends the message $U$ with rate $R$. Let the compression loss be $D$, then: $R + D \geq I(X;Y)$*

*Proof.*

$$\begin{aligned}
R + D &= H(U) + D \\
&= H(U) + I(X;Y|U) + D\left(p(y|u)||\hat{p}(y|u)\right) \\
&= H(U) + H(X|U) - H(X|Y,U) + D\left(p(y|u)||\hat{p}(y|u)\right) \\
&= H(X,U) - H(X|Y) + H(X|Y) - H(X|Y,U) + D\left(p(y|u)||\hat{p}(y|u)\right) \\
&= H(X) - H(X|Y) + H(U|X) + I(X;U|Y) + D\left(p(y|u)||\hat{p}(y|u)\right) \\
&= I(X;Y) + H(U|X) + I(X;U|Y) + D\left(p(y|u)||\hat{p}(y|u)\right) \\
&\geq I(X;Y)
\end{aligned}$$

There are a few points to note here:

1. One is that, this argument also hold if we are a single message $U$ for samples $X^n$ and compression $Y^n$. We will still have $R + D \geq I(X;Y)$, where the rate is defined as $\frac{H(U)}{n}$, and the compression loss is also per symbol.
2. The second point is that for optimality of this relationship, all of the remaining terms needs to be zero, which as we will see gives good characterization for the rate region.
3. The third point to note is that this analysis gives us a lower bound on the compression loss whenever we use some rate R. This analysis should hopefully be useful when we extend the observations to the recurrent neural network problem.But, also note that the bound might not be tight in any way (we will analyze the tightness of this bound in the next section)

□

The next theorem is a multiletter version of this bound.

**Theorem 2.** *If Alice sends the message $W$ for the source $X^n$, with rate $R = \frac{H(W)}{n}$. Let the compression loss be $D$ per symbol, then: $R + D \geq I(X;Y)$*

*Proof.*

$$\begin{aligned}
n(R + D) &= H(W) + D \\
&= H(W) + I(X^n;Y^n|W) + D\left(p(y^n|W)||\hat{p}(y^n|W)\right) \\
&= I(X^n;Y^n) + H(W|X^n) + I(X^n;W|Y^n) + D\left(p(y^n|u)||\hat{p}(y^n|W)\right) \\
&\geq nI(X;Y)
\end{aligned}$$

For the optimality of this multiletter $R + D$ relationship, we need: $\lim_{n\to} \frac{H(W|X^n)}{n} = 0$ and $\lim_{n\to} \frac{I(X^n;W|Y^n)}{n} = 0$. We use this observation in the next section to understand the $R, D$ region. □

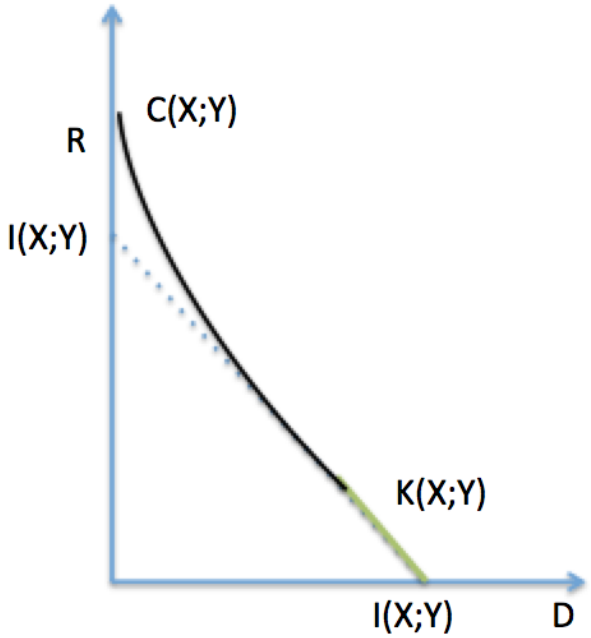


Figure 3: The $(R, D)$ rate region

## 4.1 Rate region analysis

**Lemma 3.** *$R + D = I(X;Y)$ is achievable if $R \leq K(X;Y)$ where $K(X;Y)$ is the Gacs-Korner common information.*

*Proof.* This is more of a argument (conjecture) rather than a proof, as I am yet to understand Gacs-Korner common information. The argument is based on the analysis of Theorem 2 where equaltty holds if: $\lim_{n\to} \frac{H(W|X^n)}{n} = 0$ and $\lim_{n\to} \frac{I(X^n;W|Y^n)}{n} = 0$.

The relations hold for $W = f(X^n) = g(Y^n)$ with high probability. This immediately gives us the maximum rate to be $K(X;Y)$ by the definition of the Gacs-Korner common information. □

**Lemma 4.** *$D = 0$ corresponds to $R = C(X;Y)$ which is the Wyner's common information*

*Proof.* I believe this should follow from Paul Cuff's work on distributed channel synthesis (But need to verify this perfectly) □

### 4.1.1 Analysis

The rate region is as shown in Figure 3. We present a analysis of the same below:

1. The rate region lies above the line $R + D = I(X, Y)$, which is true based on Theorem 2.
2. The D-intercept is exactly $I(X;Y)$, as if we do not send anything $(R = 0)$, we can still achieve distortion equivalent to compression $Y$ to $H(Y)$ bits, which correspond to a distortion of $H(Y) - H(Y|X) = I(X;Y)$.
3. The R-intercept point is $C(X;y)$ from Lemma 4
4. We achieve perfect relationship of $R + D = I(X;Y)$ for $R \leq K(X;Y)$. This follows because of Lemma 3, and that the region is convex.

## References


[D+15] Łukasz Dębowski et al. Hilberg's conjecture–a challenge for machine learning. *Schedae Informaticae*, 2014(Volume 23):3344, 2015.

[Dęb14] Łukasz Dębowski. On hidden markov processes with infinite excess entropy. *Journal of Theoretical Probability*, 27(2):539–551, 2014.

[Gra13] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.