Recurrent Networks

ML Instruction Team, Fall 2022

CE Department
Sharif University of Technology

Recurrent Neural Network

- A variant of the conventional feed-forward artificial neural networks to deal with sequential data
- Hold the knowledge about the past (Have memory!)
- The Unreasonable Effectiveness of Recurrent Neural Networks

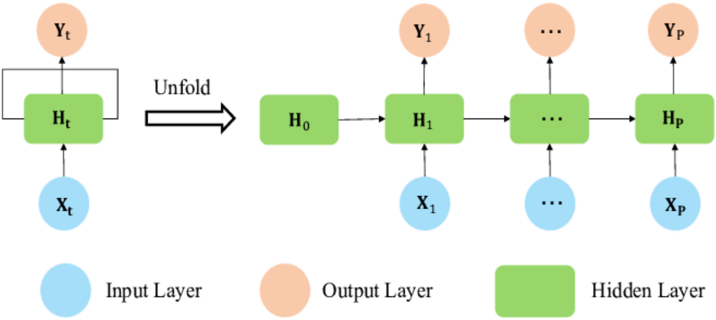


Figure: The folded and unfolded structure of recurrent neural networks, source

Fake Wikipedia Page!

```
Naturalism and decision for the majority of Arab countries' capitalide was grounded
by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated
with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal
in the [[Protestant Immineners]], which could be said to be directly in Cantonese
Communication, which followed a ceremony and set inspired prison, training. The
emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom
of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known
in western [[Scotland]], near Italy to the conquest of India with the conflict.
Copyright was the succession of independence in the slop of Syrian influence that
was a famous German movement based on a more popular servicious, non-doctrinal
and sexual power post. Many governments recognize the military housing of the
[[Civil Liberalization and Infantry Resolution 265 National Party in Hungary]],
that is sympathetic to be to the [[Punjab Resolution]]
(PJS)[http://www.humah.yahoo.com/guardian.
cfm/7754800786d17551963s89.htm Official economics Adjoint for the Nazism, Montgomery
was swear to advance to the resources for those Socialism's rule,
was starting to signing a major tripad of aid exile.]]
```

Figure: In case you were wondering, the yahoo url in the generated Wikipedia page doesn't actually exist, the model just hallucinated it.

Fake Algebraic Geometry Book!

For $\bigoplus_{n=1,\dots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ (by the formal open covering X and a single map $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ over U compatible with the complex

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If T is surjective we may assume that T is connected with residue fields of S. Moreover there exists a closed subspace $Z \subset X$ of X where U in X' is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem

(1) *f is locally of finite type. Since $S = \mathrm{Spec}(R)$ and $Y = \mathrm{Spec}(R)$.*

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\dots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{X,\dots,0}$.

Lemma 0.2. *Let X be a locally Noetherian scheme over S, $E = \mathcal{F}_{X/S}$. Set $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. Since $\mathcal{I}^n \subset \mathcal{I}^n$ are nonzero over $i_0 \le \mathfrak{p}$ is a subset of $\mathcal{J}_{n,0} \circ \overline{A}_2$ works.*

Lemma 0.3. *In Situation ??. Hence we may assume $\mathfrak{q}' = 0$.*

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. □

Figure: A sample of a recurrent network. The network is trained on the raw Latex source file of a book on algebraic geometry. Amazingly, the resulting sampled Latex almost compiles!

Fake C Code!

```c
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

Figure: This time the network is trained on the linux source code. Notice the comments, pointer notation and brackets in the above code. What are the code errors?

The Effectiveness of Recurrent Neural Networks

- All previous examples were generated blindly by recurrent neural network with simple architectures.
- Interested? Take a look at the source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Modeling Series

- In many situations one must consider a series of inputs to produce an output.
  - Outputs too may be a series
- Examples...?

Example 1: Speech Recognition

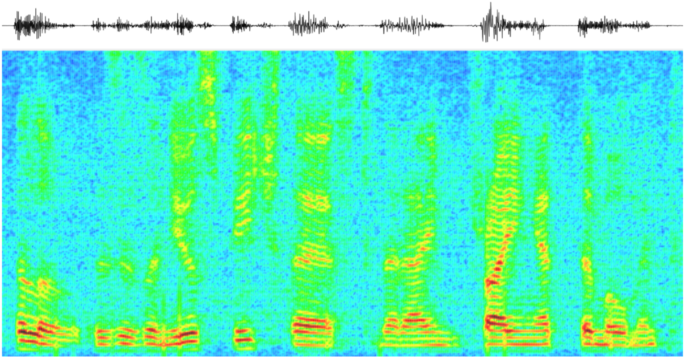


Figure: source

- Speech Recognition
  - Analyze a series of spectral vectors, determine what was said.
- Note: Inputs are sequences of vectors. Output is a classification result.

Example 2: Text Analysis

Stephen Curry scored 34 points and was named the NBA Finals MVP as the Warriors claimed the franchise's seventh championship overall. And this one completed a journey like none other, after a run of five consecutive finals, then a plummet to the bottom of the NBA, and now a return to greatness just two seasons after having the league's worst record.

- Football or Basketball?
- Text Analysis
  - E.g. analyze document, identify topic
    - Input series of words, output classification output
  - E.g. read English, output Persian
    - Input series of words, output series of words

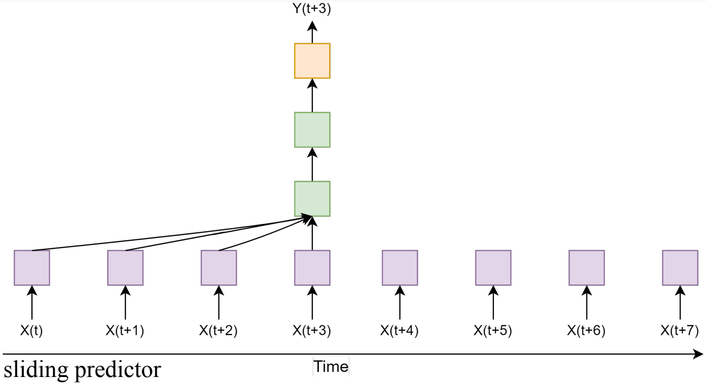
Example 3: Stock Market Prediction

- Stock Market Prediction
  - Should I invest, vs. should I not invest in X?
  - Decision must be taken considering how things have fared over time.
- Note: Inputs are sequences of vectors. Output may be scalar or vector.

Stock Market Prediction

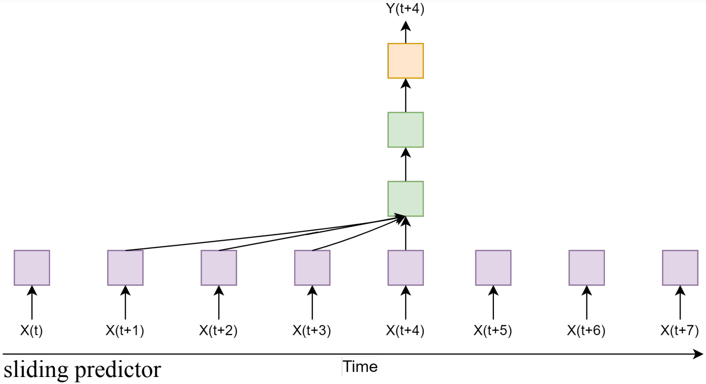
- Stock Market Prediction
  - Must consider the series of stock values several days in the past to decide
  - How should we design our network?

The Stock Predictor Network



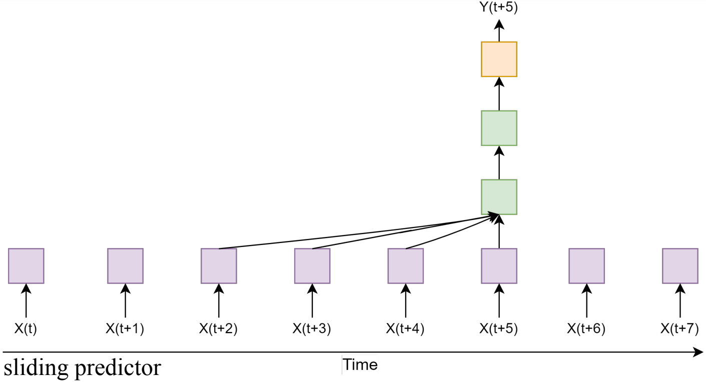
- The sliding predictor
  - Look at the last few days
  - This is just an CNN applied to series data
    - Also called a Time-Delay neural network
- Representational shortcut
  - Input at each time is a vector
  - Each box actually represents an entire layer with many units

The Stock Predictor Network



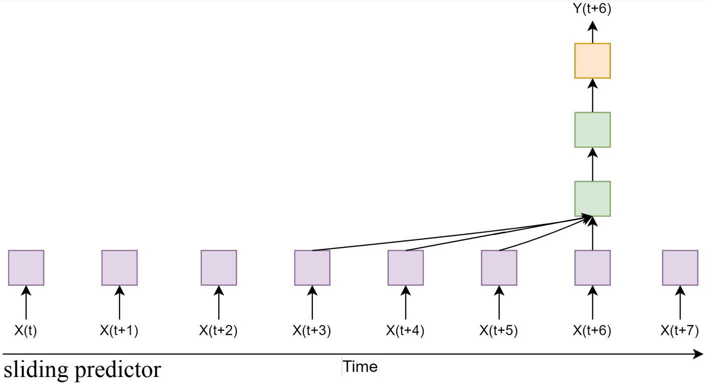
- The sliding predictor
  - Look at the last few days
  - This is just an CNN applied to series data
    - Also called a Time-Delay neural network
- Representational shortcut
  - Input at each time is a vector
  - Each box actually represents an entire layer with many units

The Stock Predictor Network



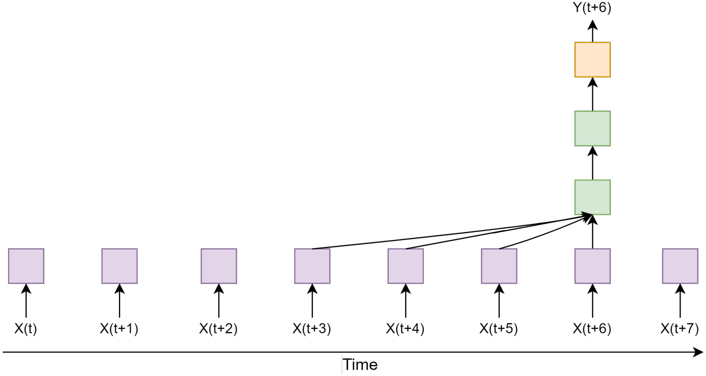
- The sliding predictor
  - Look at the last few days
  - This is just an CNN applied to series data
    - Also called a Time-Delay neural network
- Representational shortcut
  - Input at each time is a vector
  - Each box actually represents an entire layer with many units

The Stock Predictor Network



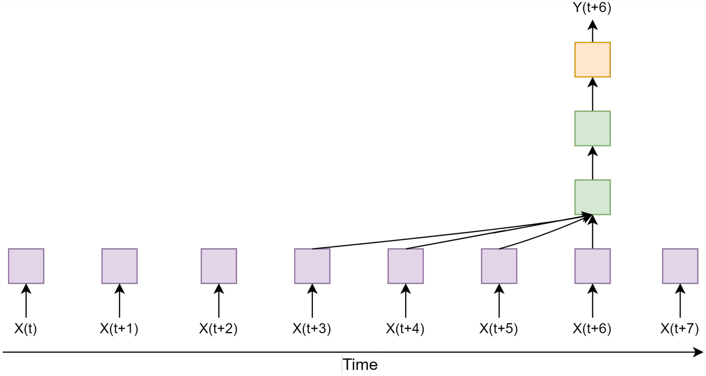
- The sliding predictor
  - Look at the last few days
  - This is just an CNN applied to series data
    - Also called a Time-Delay neural network
- Representational shortcut
  - Input at each time is a vector
  - Each box actually represents an entire layer with many units

Finite Response Model



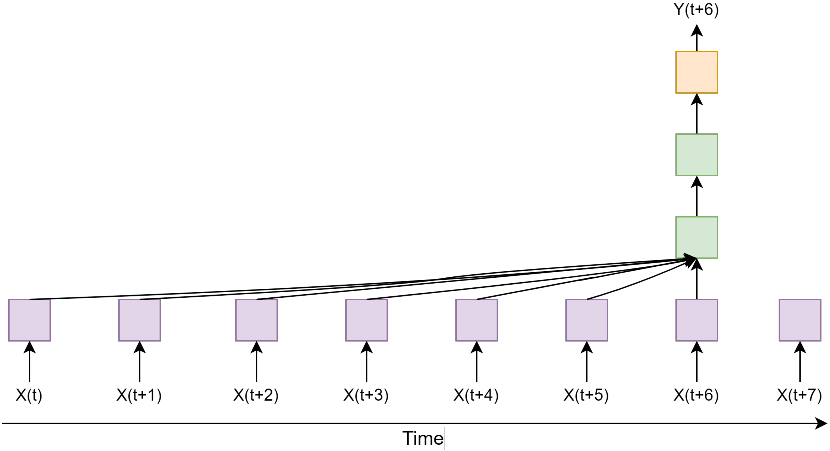
- This is a finite response system
  - Something that happens today only affects the output of the system for N days into the future
    - N is the width of the system
  - $Y_t = f(X_t, X_{t-1}, ..., X_{t-N})$

Finite Response Model



- Something that happens today only affects the output of the system for days into the future
  - Predictions consider N days of history
- What if we need to consider more of the past to make predictions?
  - Increase the “history”

Finite Response Model



- Problem: Increasing the “history” makes the network more complex

Long-Term Dependencies

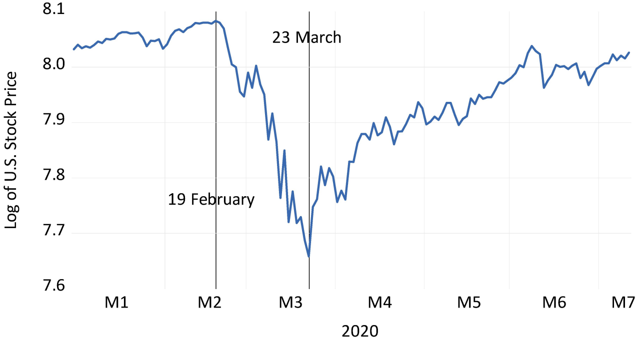
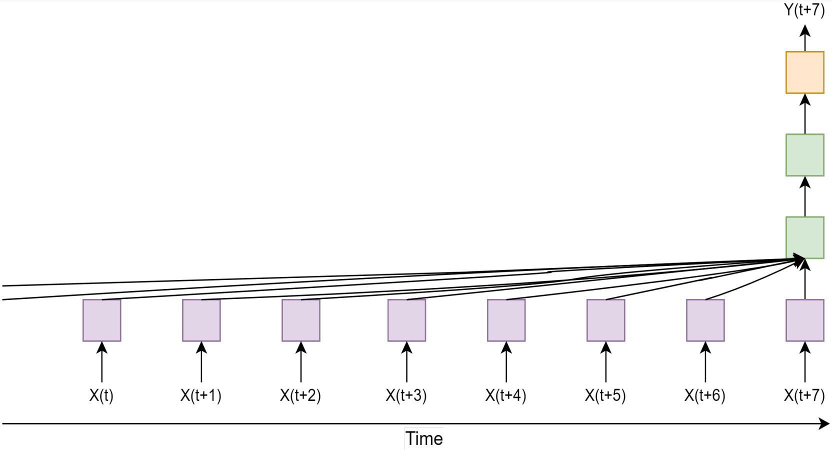


Figure: The Impact of the COVID-19 Pandemic on the U.S. Economy, source

- Systems often have long-term dependencies
  - Weekly/Monthly/Annual trends in the market
  - Though longer historic events tends to affect us less than more recent events
- Can you think of an example?

Infinite Memory



- Infinite response systems
  - What happens today can continue to affect the output forever
    - Possibly with weaker and weaker influence
  - $Y_t = f(X_t, X_{t-1}, ..., X_{t-\infty})$

RNN, An Infinite Response System

- We can process a sequence of vectors x by applying a recurrence formula at every time step
  - $h_t = f(x_t, h_{t-1}), \qquad y_t = g(h_t)$
  - h_t is the state of the network
  - x_t is the input vector at t
  - y_t is the output at t
  - Need to define initial state h_{-1} for $t = 0$
  - An input x_0 at $t = 0$ produces h_0
  - h_0 produces h_1 which produces h_2 and so on...
  - h_t can be produced from h_{t-1} even if x_t is 0
  - A single input influences the output for the rest of time
- This is a fully recurrent neural network, or simply a recurrent neural network
- Don't worry, we will get back to this slide

Vanilla Neural Networks

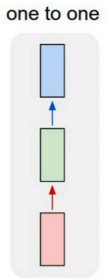


Figure: Types of Sequence Problems, source

- Vanilla Neural Networks
- Example: Image Classification
- Fixed-sized input and output

Sequence Output

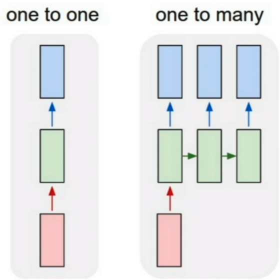


Figure: Types of Sequence Problems, source

- Sequence Output
- Example: Image Captioning
- image → sequence of words

Sequence Input

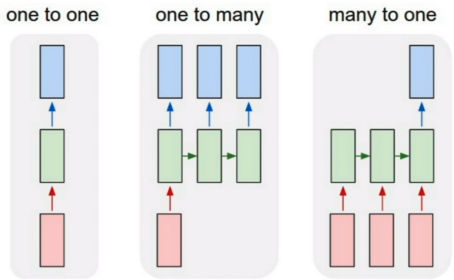


Figure: Types of Sequence Problems, source

- Sequence Input
- Example: Sentiment Analysis
- sequence of words → sentiment

Sequence Input And Sequence Output

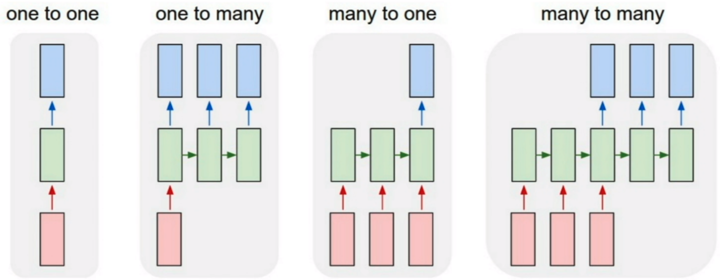


Figure: Types of Sequence Problems, source

- Sequence Input And Sequence Output
- Example: Machine Translation
- sequence of words in English → sequence of words in Persian

Synced Sequence Input And Output

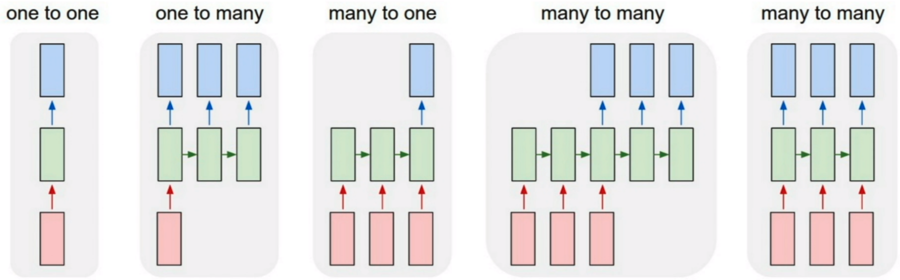


Figure: Types of Sequence Problems, source

- Synced Sequence Input And Output
- Example: Video Classification
- frames of the video → label of each frame

Latent Variable Model

- In n-grams for language modeling the conditional probability of token x_t at time step t only depends on the n previous tokens.
- If we want to incorporate the possible effect of tokens earlier than time step $t-n$ on x_t, we need to increase n
- By increasing n the number of model parameters would also increase exponentially with it
- Hence, rather than modeling $\mathbb{P}(x_t|x_{t-1},\dots,x_{t-n})$ it is preferable to use a latent variable model:

$$\mathbb{P}(x_t|x_{t-1},\dots,x_1) \approx \mathbb{P}(x_t|h_{t-1})$$

- h_{t-1} is a hidden state that stores the sequence information up to time step $t-1$
- In general, the hidden state at any time step t could be computed based on both the current input x_t and the previous hidden state h_{t-1} :

$$h_t = f(x_t, h_{t-1})$$

Recurrent Neural Network

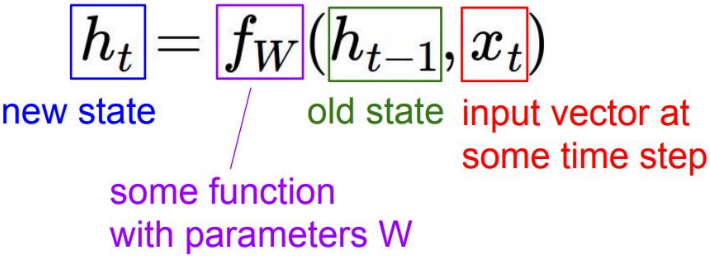
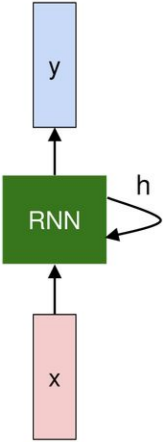


Figure: RNN formula, source

- We can process a sequence of vectors x by applying a recurrence formula at every time step
- The same function and the same set of parameters are used at every time step.

Vanilla RNN



$$h_t = f_W(h_{t-1}, x_t)$$
$$y_t = g_W(h_t)$$

$$\rightarrow \begin{cases} h_t = \mathit{tanh}(W_{hh}h_{t-1} + W_{hx}x_t + b_h) \\ y_t = W_{yh}h_t + b_y \end{cases}$$

RNN: Forward Pass

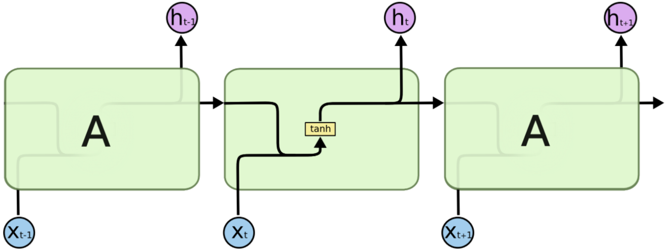


Figure: The repeating module in a standard RNN contains a single layer, source

$$
\begin{aligned}
h_t &= tanh(W_{hh}h_{t-1} + W_{hx}x_t + b_h) \\
&= tanh((W_{hh}W_{hx})\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} + b_h) \\
&= tanh(W\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} + b_h)
\end{aligned}
$$

RNN: Computational Graph

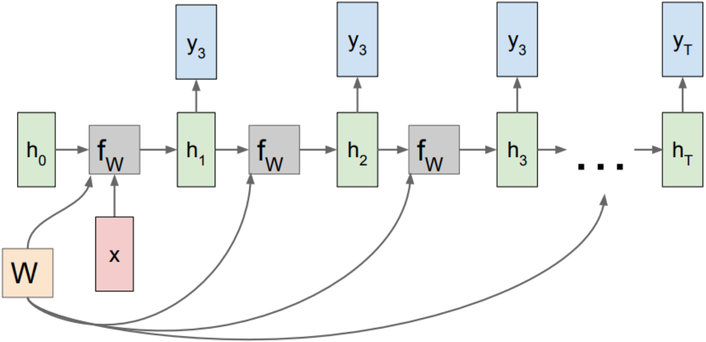


Figure: RNN One to Many Computational Graph, source

RNN: Computational Graph

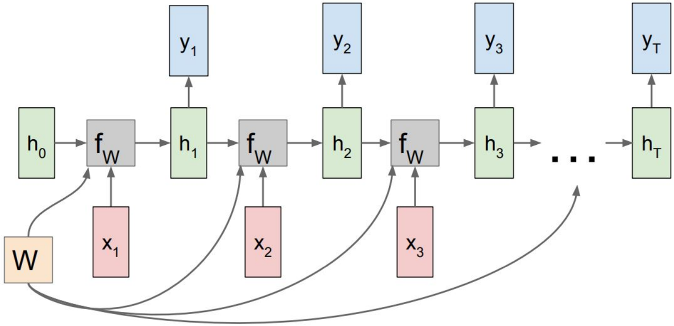


Figure: RNN Many to Many Computational Graph, source

Example: Character-Level Language Model

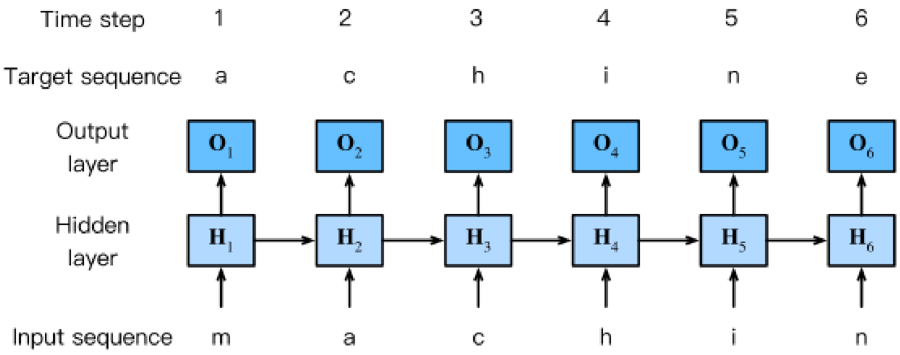
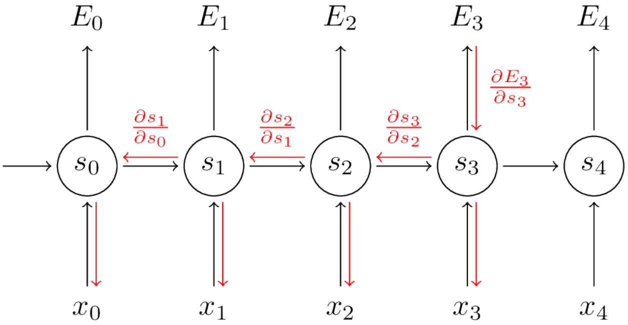


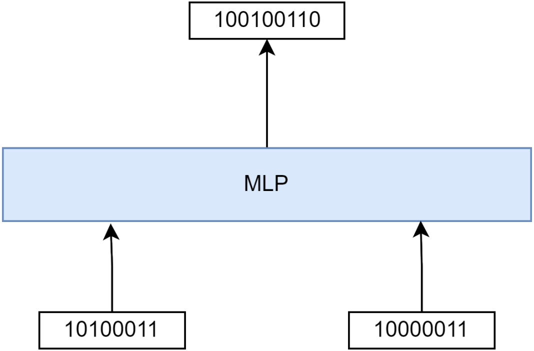
Figure: A character-level language model based on the RNN. The input and target sequences are “machin” and “achine”, respectively, source

Training RNN: Backpropagation Through Time



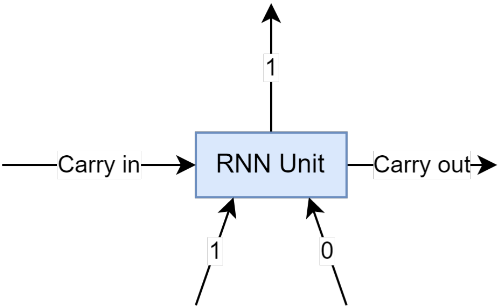
- We will explain BPTT fully in the next session

MLPs vs RNN: The Addition Problem



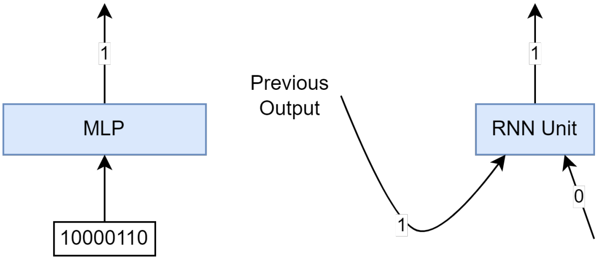
- The addition problem: Add two N-bit numbers to produce a N+1-bit number
  - Input is binary
  - MLP will require large number of training instances
  - Network trained for N-bit numbers will not work for N+1 bit numbers

MLPs vs RNN: The Addition Problem



- The addition problem: Add two N-bit numbers to produce a N+1-bit number
  - RNN solution: Very simple
  - Can add two numbers of any size
  - Needs very little training data

MLPs vs RNN: The Parity Problem



- Is the number of “ones” even or odd
  - MLP solution: XOR network, quite complex
  - RNN solution: Simple, generalizes to input of any size

Thank You!

Any Question?