Lecture 6: Recurrent Neural Networks

Alireza Akhavan Pour

CLASS.VISION
Motivating example:

X: Harry Potter and Hermione Granger invented a new spell

\[
\begin{matrix}
x^{<1>} & x^{<2>} & x^{<3>} & x^{<4>} & x^{<5>} & x^{<6>} & x^{<7>} & x^{<8>} & x^{<9>}
\end{matrix}
\]

Y: 1 1 0 1 1 0 0 0 0

\[
\begin{matrix}
y^{<1>} & y^{<2>} & y^{<3>} & y^{<4>} & y^{<5>} & y^{<6>} & y^{<7>} & y^{<8>} & y^{<9>}
\end{matrix}
\]

• Both elements has a shape of 9.
• 1 means its a name, while 0 means its not a name.
Notation

- We will index the first element of \( x \) by \( x^{<1>} \), the second \( x^{<2>} \) and so on.
  - \( x^{<1>} = \) Harry
  - \( x^{<2>} = \) Potter

- Similarly, we will index the first element of \( y \) by \( y^{<1>} \), the second \( y^{<2>} \) and so on.
  - \( y^{<1>} = 1 \)
  - \( y^{<2>} = 1 \)

- \( T_x \) is the size of the input sequence and \( T_y \) is the size of the output sequence.
  - \( T_x = T_y = 9 \) in the last example although they can be different in other problems.

- \( x^{(i)<t>} \) is the element t of the sequence of input vector \( i \). Similarly \( y^{(i)<t>} \) means the t-th element in the output sequence of the i training example.

- \( T_x^{(i)} \) the input sequence length for training example \( i \). It can be different across the examples. Similarly for \( T_y^{(i)} \) will be the length of the output sequence in the i-th training example.
While converting, what if we meet a word that's not in your dictionary?

- We can add a token in the vocabulary with name `<UNK>` which stands for unknown text and use its index for your one-hot vector.
Why not a standard network?

Problems:
- Inputs, outputs can be different lengths in different examples.
- Doesn't share features learned across different positions of text/sequence.
Forward propagation

\[
\begin{align*}
a^{<0>} &= \rightarrow_0 \\
a^{<1>} &= g_1(w_{aa}a^{<0>} + w_{ax}x^{<1>} + b_a) \quad \text{tanh, Relu} \\
\hat{y}^{<1>} &= g_2(w_{ya}a^{<1>} + b_y) \quad \text{sigmoid, Softmax} \\
a^{<t>} &= g_1(w_{aa}\,a^{<t-1>} + w_{ax}x^{<t>} + b_a) \\
\hat{y}^{<t>} &= g_2(w_{ya}a^{<t>} + b_y)
\end{align*}
\]
Simplified RNN notation

\[ a^{<t>} = g_1(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a) \]

\[ \hat{y}^{<t>} = g_2(w_{ya}a^{<t>} + b_y) \]

بازنویسی به صورت ساده تر:

\[ a^{<t>} = g_1(w_a [a^{<t-1>}, x^{<t>}] + b_a) \]

\[ \hat{y}^{<t>} = g_2(w_ya^{<t>} + b_y) \]

\[ w_{aa} \]

\[ w_{ax} \]

\[ = w_a \]

\[ (100, 10,100) \]
Simplified RNN notation

\[ a^{<t>} = g_1(w_a [a^{<t-1>}, x^{<t>}] + b_a) \]
\[ \hat{y}^{<t>} = g_2(w_y a^{<t>} + b_y) \]

- \( w_a \) is \( w_{aa} \) and \( w_{ax} \) stacked horizontally.
- \([a^{<t-1>}, x^{<t>}]\) is \( a^{<t-1>} \) and \( x^{<t>} \) stacked vertically.
- \( w_a \) shape: \((\text{NoOfHiddenNeurons}, \text{NoOfHiddenNeurons} + n_x)\)
- \([a^{<t-1>}, x^{<t>}]\) shape: \((\text{NoOfHiddenNeurons} + n_x, 1)\)
backpropagation

\[ \mathcal{L}^{<1>} \]

\[ \hat{y}^{<1>} \]

\[ \mathcal{L}^{<2>} \]

\[ \hat{y}^{<2>} \]

\[ \mathcal{L}^{<3>} \]

\[ \hat{y}^{<3>} \]

\[ \mathcal{L}^{<T_x>} \]

\[ \hat{y}^{<T_x>} \]

\[ \mathcal{L} \]

\[ w_y, b_y \]

\[ \alpha^{<0>} \]

\[ \alpha^{<1>} \]

\[ \alpha^{<2>} \]

\[ \alpha^{<3>} \]

\[ \mathcal{L}^{<t>} (\hat{y}^{<t>}, y^{<t>}) = -y^{<t>} \log \hat{y}^{<t>} - (1 - y^{<t>}) \log(1 - \hat{y}^{<t>}) \]

\[ \mathcal{L} (\hat{y}, y) = - \sum_{t=1}^{T_y} \mathcal{L}^{<t>} (\hat{y}^{<t>}, y^{<t>}) \]
What is a language model?

speech recognition

- Pair or pear?!ˈpɜːr\ˈper\
- Pair and pear sounds exactly the same, so how would a speech recognition application choose from the two.

- The apple and pair salad.
- The apple and pear salad.

- \( P(\text{The apple and pair salad}) = 0.2 \times 10^{-10} \)
- \( P(\text{The apple and pear salad}) = 0.3 \times 10^{-5} \)
Language modelling with an RNN

✓ training set: a large corpus of target language text.
✓ tokenize
✓ Put an end of sentence token <EOS>
✓ token <UNK> for the unknown words.

Cats average 15 hours of sleep a day. <EOS>

\[ y^{<1>} \quad y^{<2>} \quad y^{<3>} \quad \ldots \quad y^{<9>} \]

The Egyptian Mau is a bread of cat. <EOS>

\[ x^{<t>} = y^{<t-1>} \quad <\text{UNK}> \]
RNN model

\[ a^{<0>} = \vec{0} \rightarrow \rightarrow a^{<1>} \]
\[ x^{<1>} = \vec{0} \rightarrow \rightarrow \] cats
\[ y^{<1>} \rightarrow \rightarrow \]

\[ a^{<2>} \rightarrow \rightarrow a^{<3>} \rightarrow \rightarrow \]
\[ x^{<2>} = y^{<1>} \rightarrow \rightarrow \] average
\[ x^{<3>} = y^{<3>} \rightarrow \rightarrow \]
\[ y^{<3>} \rightarrow \rightarrow \]

\[ a^{<9>} \rightarrow \rightarrow a^{<9>} \]
\[ x^{<9>} = y^{<8>} \rightarrow \rightarrow \] day

Cats average 15 hours of sleep a day.

\[ p(\text{average} | \text{cats}) \]

\[ L(\hat{y}^{<t>}, y^{<t>}) = - \sum_i y_i^{<t>} \log \hat{y}_i^{<t>} \]

\[ L = \sum_t L^{<t>}(\hat{y}^{<t>}, y^{<t>}) \]
1. We first pass $a^{<0>}$ and $x^{<1>}$ = zeros vector.

2. Then we choose a prediction randomly from distribution obtained by $\hat{y}^{<1>}$; For example it could be "The".
   - In numpy this can be implemented using: `numpy.random.choice(...)`
   - This is the line where you get a random beginning of the sentence each time you sample run a novel sequence.

3. We pass the last predicted word with the calculated $a^{<1>}$

4. We keep doing steps 3 & 4 for a fixed length or until we get the $<$EOS$>$ token.

5. You can reject any $<$UNK$>$ token if you mind finding it in your output.
character-level language model

In the character-level language model, the vocabulary will contain [a-zA-Z0-9], punctuation, special characters and possibly token.
Sonnet 116 – Let me not ...

by William Shakespeare

Let me not to the marriage of true minds
    Admit impediments. Love is not love
Which alters when it alteration finds,
    Or bends with the remover to remove:
O no! it is an ever-fixed mark
    That looks on tempests and is never shaken;
It is the star to every wandering bark,
    Whose worth's unknown, although his height be taken.
Love's not Time's fool, though rosy lips and cheeks
    Within his bending sickle's compass come:
Love alters not with his brief hours and weeks,
    But bears it out even to the edge of doom.
If this be error and upon me proved,
    I never writ, nor no man ever loved.
at first:

```plaintext
tyntd-iafhatawiaoihrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e plia tkrlgd t o idoe ns,smmt h ne etie h,hregtrs nigtkie,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwyl fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```
PANDARUS:
Alas, I think he shall be come approached and the day
When little strain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
My fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.

VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
open source textbook on algebraic geometry

Latex source
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 $\text{Proj}_X(A) = \text{Spec}(B)$ over $U$ compatible with the complex $\text{Set}(A) = \Gamma(X, \mathcal{O}_X, \mathcal{O}_X)$. When in this case of to show that $Q \rightarrow C_{2/X}$ is stable under the following result in the second conditions of (3), 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 $S = \text{Spec}(R)$ and $Y = \text{Spec}(R)$.

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

The following lemma surjective restrocomposes of this implies that $F_{x_0} = F_{x_0} = F_{X_{i_0}}$.

Lemma 0.2. Let $X$ be a locally Noetherian scheme over $S$, $E = F_{X/S}$. Set $I = \mathcal{I}_F \subset \mathcal{I}_G$. Since $\mathcal{I}_F \subset \mathcal{I}_G$ are nonzero over $\mathcal{I}_F \cap \mathcal{I}_G$ works.

Lemma 0.3. In Situation ???. Hence we may assume $\mathcal{I}_F \neq 0$.

Proof. We will use the property we see that $P$ is the next 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 $\delta_{n+1}$ is a scheme over $S$.

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 $X_{\text{open}}$, which gives an open subspace of $X$ and $T$ equal to $S_{\text{reg}}$, see Descent, Lemma ???. Namely, by Lemma ?? we see that $R$ is geometrically regular over $S$. 

For $\bigoplus_{m=1,...,m}$ where $L_{m-1} = 0$, hence we can find a closed scheme $H$ in $H$ and any sets $F$ on $X$, $U$ is a closed immersion of $S$, then $U \rightarrow T$ is a separated algebraic space.

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

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

and the comparably in the fibre product covering we have to prove the lemma generated by $\prod Z \times U \rightarrow V$. Consider the maps $M$ along the set of points $\text{Sch}_{/X}$ and $U \rightarrow U$ 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 $\text{Sh}(G)$ such that $\text{Spec}(R') \rightarrow S$ 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 $\mathcal{O}_{X,S'}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \rightarrow \mathcal{O}_{X,x''}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\text{GL}_{S'}(x'/S'')$ and we win.

To prove study we see that $F_{U}$ is a covering of $X'$, and $F_{\mathcal{I}}$ is an object of $F_{X/S}$ for $i > 0$ and $F_{\mathcal{I}}$ exists and let $F_{\mathcal{I}}$ be a presheaf of $\mathcal{O}_X$-modules on $C$ as a $\mathcal{I}$-module. In particular $F = U/F$ we have to show that

$$\overline{\mathcal{M}}^* = \mathcal{I} \otimes_{\text{Spec}(k)} \mathcal{O}_{S, x} \otimes_{\mathcal{O}_{S, x}} \mathcal{F}$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = (\mathcal{S}\mathcal{C}/S)^{ppf} \otimes_{\text{Spec}(k)} \mathcal{O}_{S, x} \otimes_{\mathcal{O}_{S, x}} \mathcal{F}$$

and

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

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

Proof. See discussion of sheaves of sets.
Proof. Omitted.

**Lemma 0.1.** Let $\mathcal{C}$ be a set of the construction.
Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that
\[ \mathcal{O}_{\mathcal{X}} = \mathcal{O}_{\mathcal{X}}(\mathcal{L}) \]

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\text{etale}}$, we have
\[ O_X(\mathcal{F}) = \{ \text{morphs}_1 \times_{O_X} (\mathcal{G}, \mathcal{F}) \} \]
where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules.

**Lemma 0.2.** This is an integer $\mathcal{Z}$ is injective.

Proof. See Spaces, Lemma ??.

**Lemma 0.3.** Let $S$ be a scheme. Let $X$ be a scheme and $X$ is an affine open covering. Let $U \subset X$ be a canonical and locally of finite type. Let $X$ be a scheme. Let $X$ be a scheme which is equal to the formal complex.

The following to the construction of the lemma follows.

Let $X$ be a scheme. Let $X$ be a scheme covering. Let
\[ b : X \to Y' \to Y \to Y' \times_X Y \to X. \]
be a morphism of algebraic spaces over $S$ and $Y$.

Proof. Let $X$ be a nonzero scheme of $X$. Let $X$ be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

1. $\mathcal{F}$ is an algebraic space over $S$.
2. If $X$ is an affine open covering.

Consider a common structure on $X$ and $X$ the functor $\mathcal{O}_X(U)$ which is locally of finite type.
static void do_command(struct seq_file *m, void *v)
{
    int column = 32 << (cmd[2] & 0x80);
    if (state)
        cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << 1))
            pipe = (in_use & UMXTHREAD_UNCCA) +
            ((count & 0x000000000000000f) & 0x000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x20000000);
        pipe_set_bytes(i, 0);
    }
    /* Free our user pages pointer to place camera if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_controls(offset, idx, &offset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy ");
}
/*
 * Copyright (c) 2006-2010, Intel Mobile Communications. All rights reserved.
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License version 2 as published by
 * the Free Software Foundation.
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */

#include <linux/kesec.h>
#include <linux/errno.h>
#include <linux/io.h>
#include <linux/platform_device.h>
#include <linux/multi.h>
#include <linux/ckevent.h>

#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setew.h>
#include <asm/pgproto.h>
```c
#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setew.h>
#include <asm/pgproto.h>

#define REG_PG vesa_slot_addr_pack
#define PFM_NOCOMP AFSR(0, load)
#define STACK_DDR(type) (func)

#define SWAP_ALLOCATE(nr) (e)
#define emulate_sigs() arch_get_unaligned_child()
#define access_rw(TST) asm volatile("movd %esp, %0, %3 : : ".r"(0));
if (__type & DO_READ)

static void stat_PC_SEC __read_mostly offsetof(struct seq_argsqueue, 
pC>[1]);

static void
os_prefix(unsigned long sys)
{
  #ifdef CONFIG_PREEMPT
    PUT_PARAM_RAID(2, sel) = get_state_state();
    set_pid_sum((unsigned long)state, current_state_str(),
               (unsigned long)-1->lr_full; low;
  
```
Searching for interpretable cells

[Visualizing and Understanding Recurrent Networks, Andrej Karpathy*, Justin Johnson*, Li Fei-Fei]
Searching for interpretable cells

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."
Searching for interpretable cells

Cell sensitive to position in line:
The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action—the one Kutuzov and the general mass of the army demanded—namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all—carried on by vis inertiae—pressed forward into boats and into the ice-covered water and did not surrender.

line length tracking cell
Searching for interpretable cells

```c
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask, siginfo_t *info)
{
    int sig = next_signal(pending, mask);
    if (sig) {
        if (current->notifier) {
            if (sigismember(current->notifier_mask, sig)) {
                if (current->notifier)(current->notifier_data)
                clear_thread_flag(TIF_SIGPENDING);
                return 0;
            }
        }
        collect_signal(sig, pending, info);
    }
    return sig;
}
```

if statement cell
Searching for interpretable cells

```c
/* Duplicate LSM field information. The lsm_rule is opaque, so
   re-initialized. */
static inline int audit_dupe_lsm_field(struct audit_field *df,
    struct audit_field *sf)
{
    int ret = 0;
    char *lsm_str;
    /* our own copy of lsm_str */
    lsm_str = kstrdup(sf->lsm_str, GFP_KERNEL);
    if (unlikely(!lsm_str))
        return -ENOMEM;
    df->lsm_str = lsm_str;
    /* our own (refreshed) copy of lsm_rule */
    ret = security_audit_rule_init(df->type, df->op, df->lsm_str,
        (void **) &df->lsm_rule);
    /* Keep currently invalid fields around in case they
       become valid after a policy reload. */
    if (ret == -EINVAL) {
        pr_warn("audit rule for LSM \%s\ is invalid\n",
            df->lsm_str);
    ret = 0;
    }
    return ret;
}
```
Searching for interpretable cells

code depth cell
Vanishing gradients with RNNs

- "The cat, which already ate ..., was full"
- "The cats, which already ate ..., were full"

به ایران سفر کرده بودم، ..... 
در انتهای سفر زبان فارسی را به خوبی یاد گرفته بودم.
منابع

• https://www.coursera.org/specializations/deep-learning
• http://cs231n.stanford.edu/
• https://medium.com/@athif.shaffy/one-hot-encoding-of-text-b69124bef0a7