

Основы практического использования нейронных сетей.

Лекция 9. Рекуррентные нейронные сети.

Дмитрий Буряк.
к.ф.-м.н.
dyb04@yandex.ru

Решаемые задач

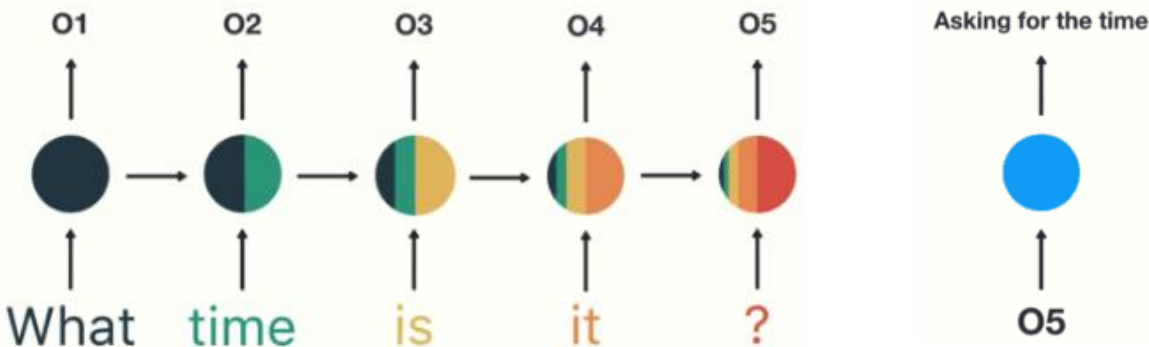
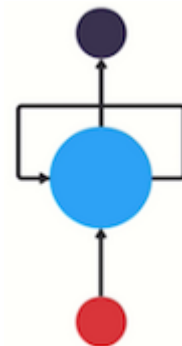
Обработка данных, составленных в последовательности.

- ☐ Распознавание речи (речь-в-текст)
- ☐ Генерация речи (текст-в-речь)
- ☐ Перевод с одного языка на другой
- ☐ Предсказание биржевых индексов
- ☐ Составление описаний изображений (изображение-в-текст)

A young boy is playing basketball. 	Two dogs play in the grass. 	A dog swims in the water. 	A little girl in a pink shirt is swinging. 
A group of people walking down a street. 	A group of women dressed in formal attire. 	Two children play in the water. 	A dog jumps over a hurdle. 

Структура рекуррентного нейрона

- ❑ Введение обратной связи для передачи информации о предшествующих входах (состояние нейрона).
- ❑ Соответствует многослойному персептрону, каждый слой имеет собственный вход, разделяемые веса.
- ❑ Вычисление признакового представления входной последовательности (рекуррентные слои) → классификация (персептрон)
- ❑ Обучение – back-propagation through time.



Back-propagation through time

- ❑ Дано: последовательности входов и заданных выходов сети.
- ❑ Шаг алгоритма BPTT:
 - «Развернуть» сеть и вычислить последовательность выходов
 - Вычислить функцию ошибку для всех выходов
 - Обновить веса, выполнив обратный проход.
- ❑ Вычислительно сложный для длинных последовательностей.
- ❑ Шаг алгоритма Truncated BPTT, входная последовательность длины $K1$:
 - «Развернуть» сеть и вычислить последовательность выходов для $K2$ входов
 - Вычислить функцию ошибку для всех выходов
 - Обновить веса, выполнив обратный проход.
- ❑ Подбор значений $K1$ и $K2$.

Проблема исчезающего градиента

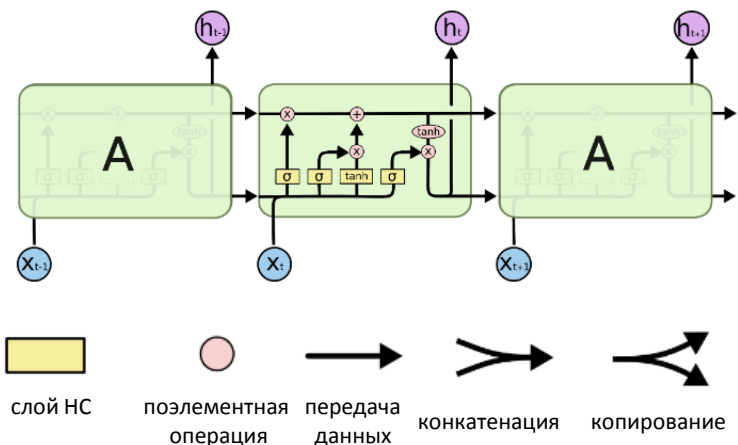
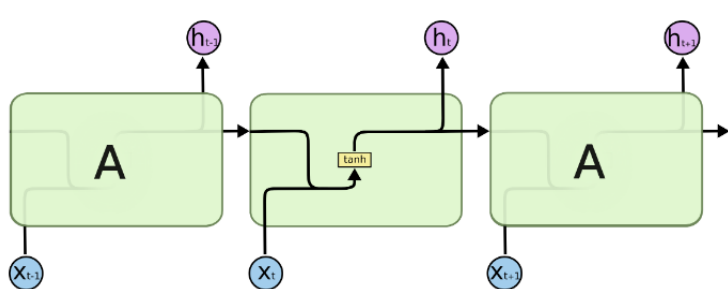
- ❑ Уменьшение значения градиента от слоя к слою.
- ❑ Проблема для многослойных сетей → для рекуррентных тоже.
- ❑ Ранние слои отвечают за обработку начальных элементов последовательности → малые значения градиента, небольшая коррекция весов.
- ❑ «Забывание» долгосрочных зависимостей.



- Влияние первых слов в последовательности на выход последнего нейрона стремится к 0

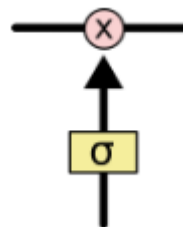
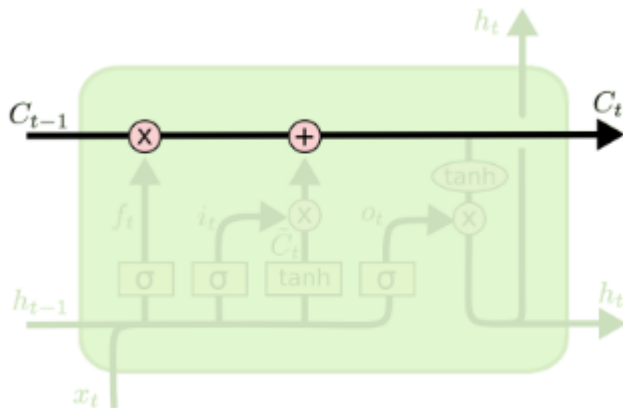
LSTM нейрон

- ❑ LSTM – Long Short Term Memory.
- ❑ Разработан для запоминания долгосрочных зависимостей.



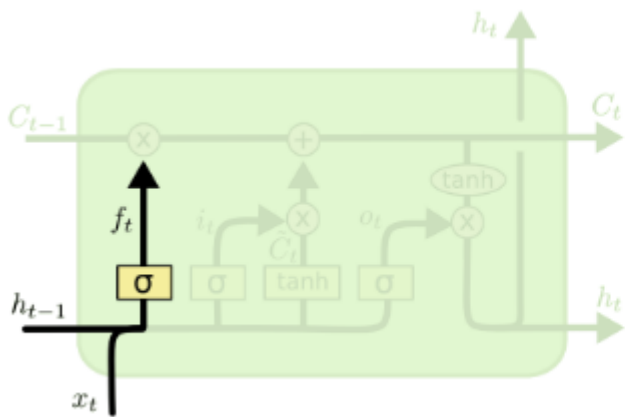
Состояние и вентили (gates)

- ❑ Вектор состояния сохраняет контекст (временные зависимости).
- ❑ Gate (вентиль) контролирует добавление и удаление информации из вектора состояния.
- ❑ Выход вентиля формируется сигмоидой.
- ❑ Типы вентилей: добавление информации (input gate), забывание информации (forget gate), формирование выхода LSTM нейрона (output gate).



Forget gate

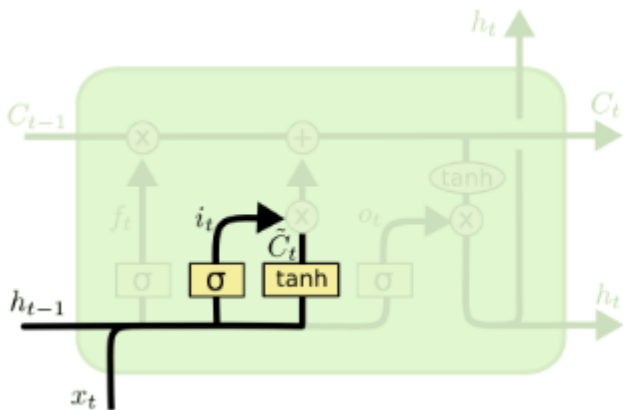
❑ Forget gate – вентиль для контроля «забывания» информации в векторе состояний.



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

Input gate

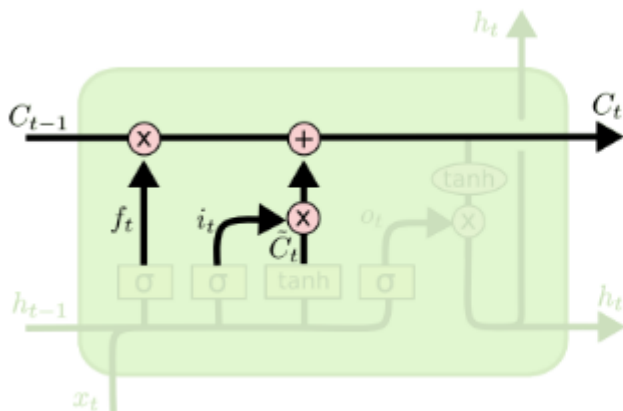
□ Input gate – вентиль для определения, в каком виде входная информация в текущий момент времени войдет в вектор состояния.



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

Обновление вектора состояния

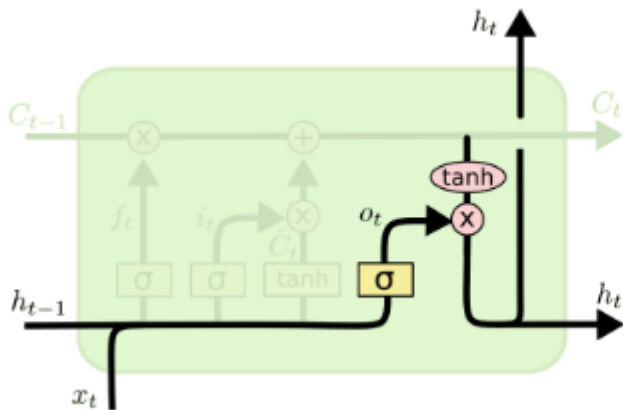
□ Вектор состояний в текущий момент времени – линейная комбинация его предыдущего варианта и нового кандидата, полученного по входной информации.



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

Выходное значение

□ Выходное значение определяется выходом традиционного полносвязного слоя, умноженного на преобразованный вектор состояний.

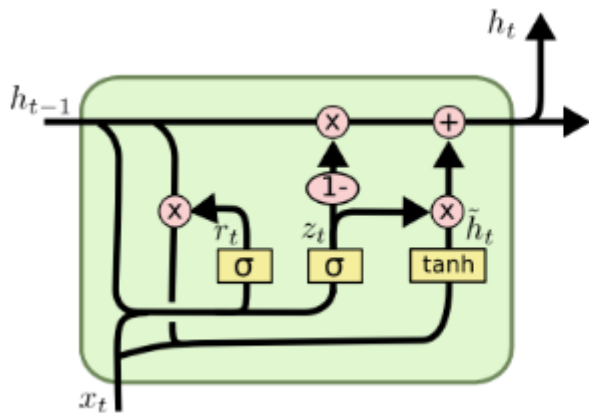


$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Gate Recurrent Unit

- ❑ Gate Recurrent Unit (GRU) – упрощенный вариант LSTM нейрона.
- ❑ Уменьшено число вентиляей.
- ❑ Совмещение вектора состояний и выхода нейрона.



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

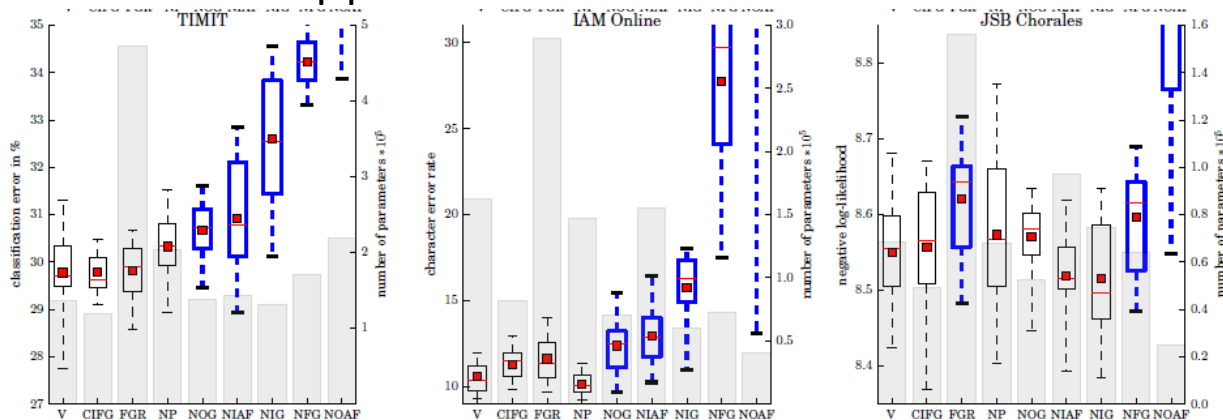
Сравнение рекуррентных нейронов

❑ Сравнение различных конфигураций LSTM нейрона: отключение вентиляй, исключение нелинейных функций активации, отказ от поэлементного умножения.

Одна из конфигураций реализует GRU (CIFG).

❑ 3 тестовые задачи: распознавание речи (TIMIT), распознавание рукописных предложений (IAM online), предсказание мелодии (JSB Chorales, Бах).

❑ Ни одна из конфигураций не является абсолютно лучшей. Ряд вариантов вычислительно более эффективные.



Пример построения

- ❑ Задача: посимвольная генерация слов.

Алфавит: 4 буквы (h, e, l, o).

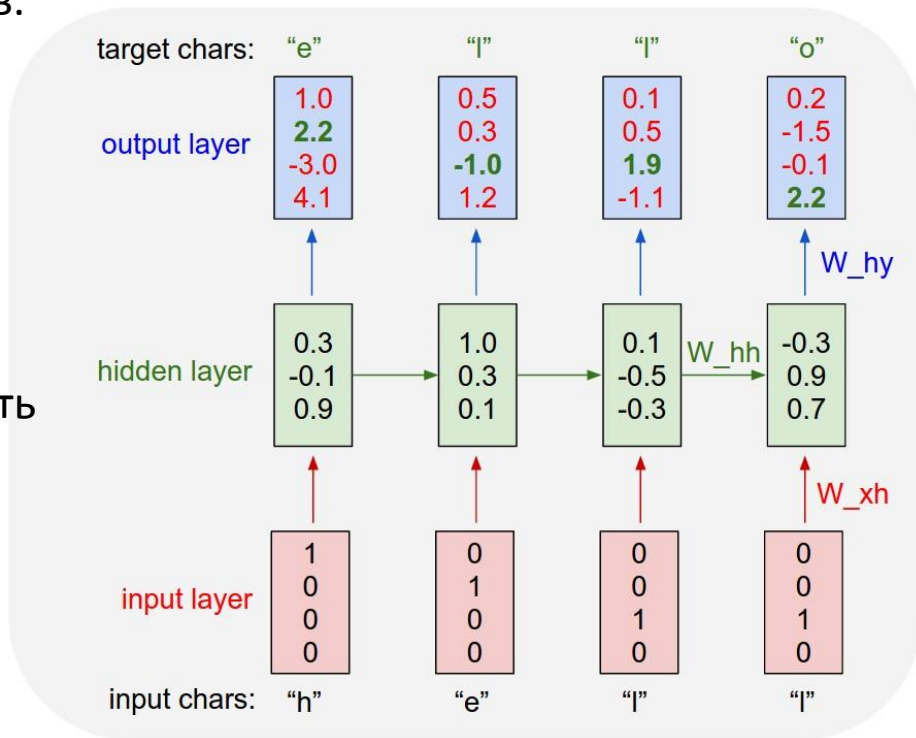
- ❑ Архитектура:

Вход – 4-мерный бинарный вектор (1-из-k).

Внутренний слой – 3 рекуррентных нейрона

Выход – 4-мерный вектор (уверенность появления буквы).

- ❑ Обучение: генерация слова “hello”



Примеры генераторов текста

- ❑ Задача: генерация текста

- ❑ Архитектура:

 - Вход – текущее слово: бинарный вектор (1-из-k).

 - Внутренний слой – 3 слоя по 512 LSTM нейронов

 - Выход – вектор степени уверенности появления слова из словаря.

- ❑ Обучение: набор текстов.

- ❑ Примеры генераторов:

 - произведения Шекспира

 - учебник по алгебраической геометрии

 - исходный код Линукса

Генератор «произведений» Шекспира

❑ Обучающая выборка – все произведения Шекспира (4.4Мб)

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 nudes 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 apt, 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.

Генератор «алгебраическая геометрия»

□ Обучающая выборка – учебник по алгебраической геометрии в Latex.

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

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \rightarrow \mathcal{F}$ of $\mathcal{O}$ -modules. □

Lemma 0.2. *This is an integer $\mathbb{Z}$ 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 $\mathcal{U} \subset X$ 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

$$b : X \rightarrow Y' \rightarrow Y \rightarrow Y \rightarrow Y' \times_X Y \rightarrow X.$$

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. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram

$$\begin{array}{ccc} S & \xrightarrow{\quad} & \\ \downarrow & & \downarrow \\ \xi & \xrightarrow{\quad} & \mathcal{O}_{X'} \\ & \uparrow & \searrow \\ & \text{got}_s & \\ & & \\ & = \alpha' & \xrightarrow{\quad} \\ & \updownarrow & \\ & = \alpha' & \xrightarrow{\quad} \alpha \end{array} \quad \begin{array}{ccc} & & X \\ & & \downarrow \\ \text{Spec}(K_\psi) & \xrightarrow{\quad} & \text{Mor}_{\text{Sets}} \text{d}(\mathcal{O}_{X_{X/\mathbb{A}}}, \mathcal{G}) \end{array}$$

is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_* . This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \text{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U . □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.

A reduced above we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a “field

$$\mathcal{O}_{X,x} \rightarrow \mathcal{F}_x \rightarrow \mathcal{O}_{X_{\acute{e}tale}}^* \rightarrow \mathcal{O}_{X_t}^* \mathcal{O}_{X_\lambda}(\mathcal{O}_{X_q}^*)$$

is an isomorphism of covering of $\mathcal{O}_{X_t}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.

The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$ -algebra with $\mathcal{F}$ are opens of finite type over S .

If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??.

This is a sequence of $\mathcal{F}$ is a similar morphism.

Генератор «С код»

❑ Обучающая выборка - исходный С код Линукса (474Мб).

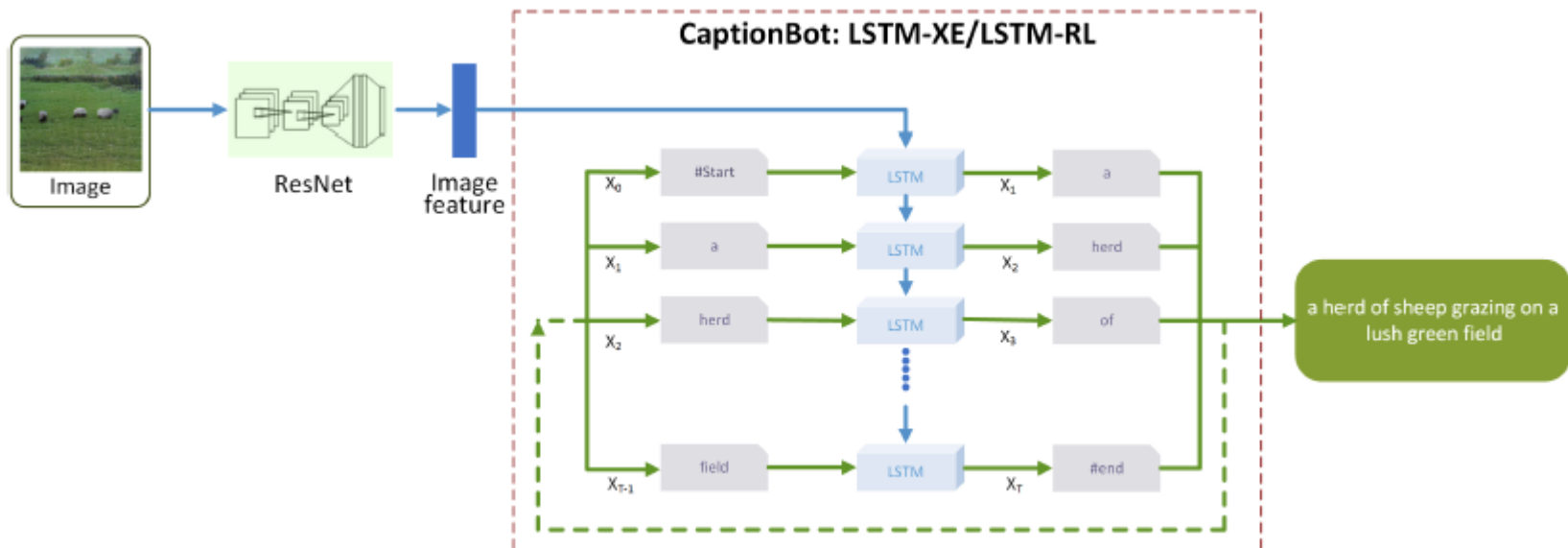
```
static void num_serial_settings(struct tty_struct *tty)
{
    if (tty == tty)
        disable_single_st_p(dev);
    pci_disable_spool(port);
    return 0;
}

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 & UMXTTHREAD_UNCCA) +
                ((count & 0x00000000ffffffff) & 0x0000000f) << 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 ");
}
```

```
/*
 * 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;
}
```

Составление описания изображения

- ❑ Задача: генерация словесного описания изображения (Image captioning).
- ❑ Вход: изображение
- ❑ Выход: вектор «вероятностей» появления каждого слова из словаря



Вопросы

- ❑ Одна из основных причин низкой эффективности обычных рекуррентных сетей?
- ❑ Основные элементы LSTM нейрона?
- ❑ Отличие GRU нейрона от LSTM нейрона?