Търсене и извличане на информация. Приложение на дълбоко машинно обучение

Стоян Михов

Лекция 12: Архитектури на рекурентни невронни мрежи с портали

План на лекцията

1. **Формалности за курса (3 мин)**
2. Особености при обучение на рекурентна невронна мрежа (30 мин)
3. Проблем и решение при експлодиращ градиент (10 мин)
4. Проблем при изчезващ градиент (10 мин)
5. Архитектури за решаване на проблема изчезващ градиент (20 мин)
6. Двупосочни и многослойни архитектури с рекурентни невронни мрежи (10 мин)
7. Приложения на рекурентните невронни мрежи за езиков модел и класификация на документи (10 мин)

Формалности

- Засега ще провеждаме занятията онлайн всяка сряда от 8:15 до 12:00 часа.
- Засега ще използваме платформата Google meet: meet.google.com/hue-frfx-axb
- Днес ще използваме едновременно слайдове и бяла дъска. Моля следете съответния екран.
- До края на тази седмица в Moodle ще бъдат публикувани оценките Домашно задание №2
- Домашното задание №3 ще бъде публикувано следващата седмица.
- Лекция 12 се базира на глава 15 от втория учебник.

План на лекцията

1. Формалности за курса (3 мин)
2. **Особености при обучение на рекурентна невронна мрежа (30 мин)**
3. Проблем и решение при експлодиращ градиент (10 мин)
4. Проблем при изчезващ градиент (10 мин)
5. Архитектури за решаване на проблема изчезващ градиент (20 мин)
6. Двупосочни и многослойни архитектури с рекурентни невронни мрежи (10 мин)
7. Приложения на рекурентните невронни мрежи за езиков модел и класификация на документи (10 мин)

Рекурентни невронни мрежи

$$\mathbf{y}_i = \text{softmax}(U\mathbf{h}_i)$$

$$\mathbf{h}_i = g(W\mathbf{h}_{i-1} + V\mathbf{x}_i)$$

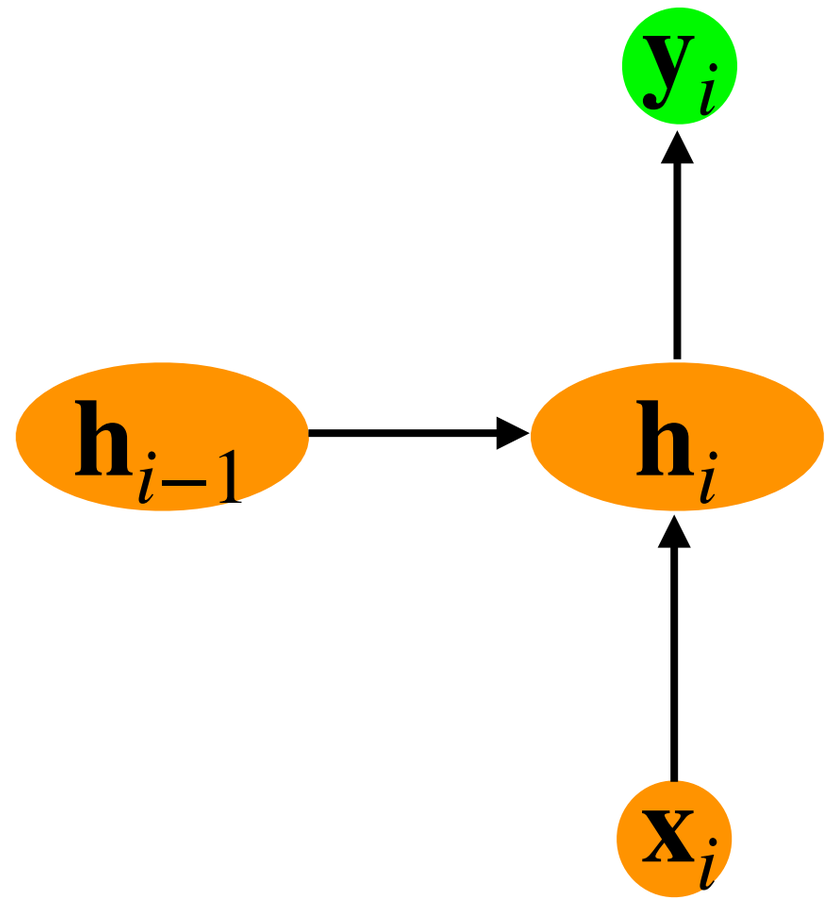
$$\mathbf{x}_i = E\chi_{w_i}$$

$$\chi_{w_i} \in \mathbb{R}^{|L|}, E \in \mathbb{R}^{M \times |L|},$$

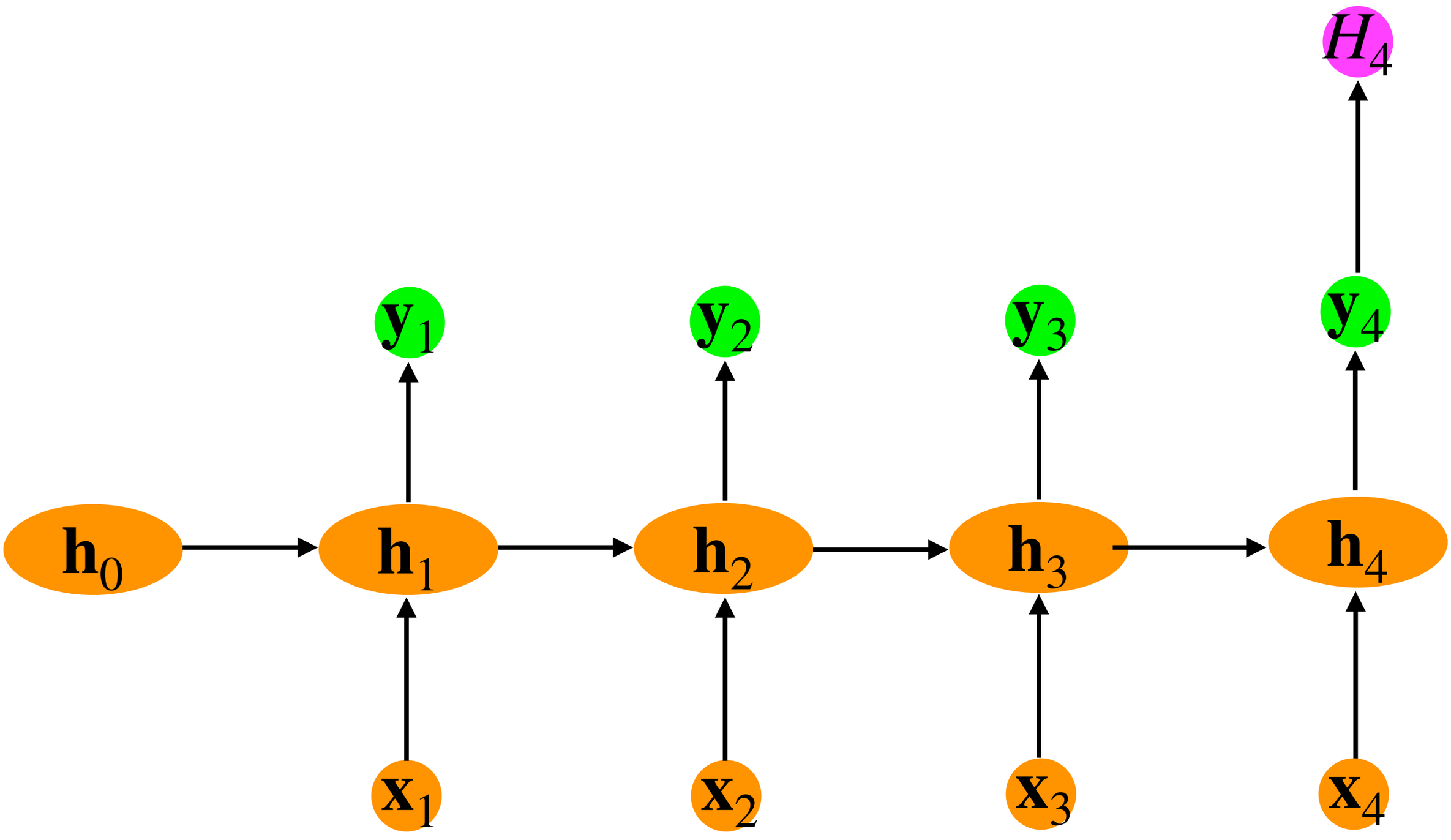
$$\mathbf{x}_i \in \mathbb{R}^M, V \in \mathbb{R}^{N \times M},$$

$$\mathbf{h}_i, \mathbf{h}_{i-1} \in \mathbb{R}^N, W \in \mathbb{R}^{N \times N},$$

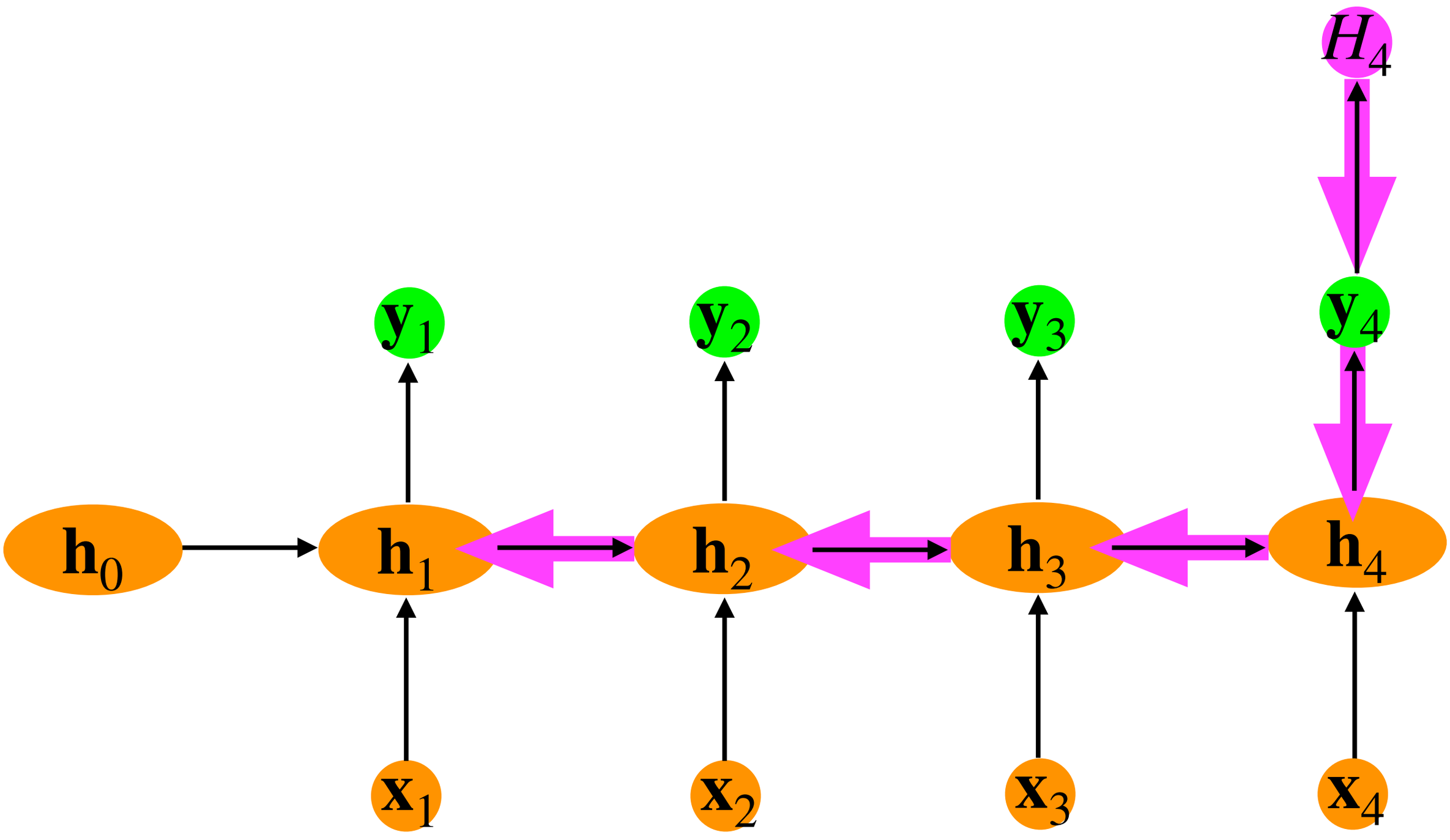
$$U \in \mathbb{R}^{|L| \times N}, \mathbf{y}_i \in \mathbb{R}^{|L|}$$



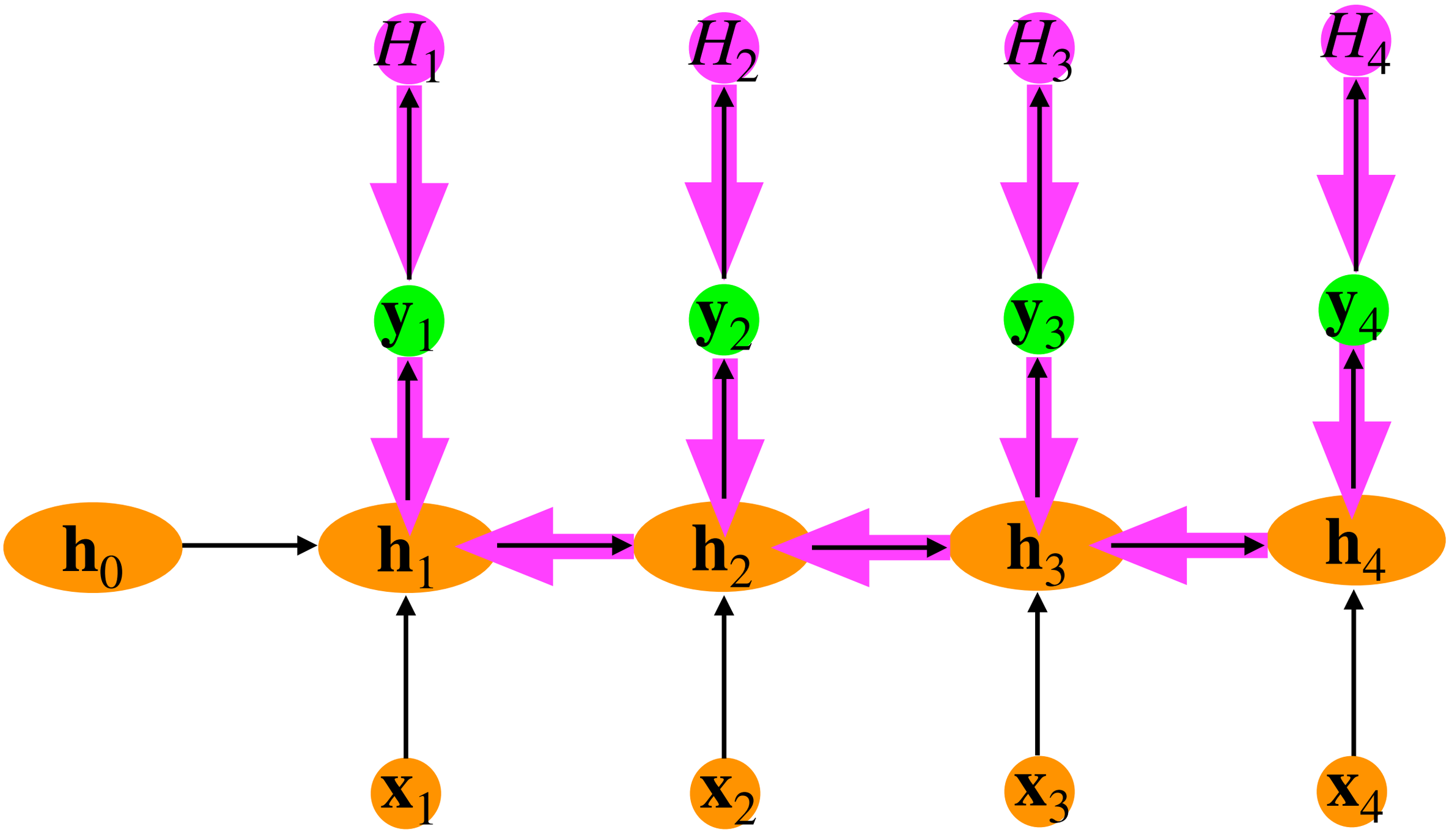
Пропагиране при рекурентни невронни мрежи



Пропагиране при рекурентни невронни мрежи



Пропагиране при рекурентни невронни мрежи



- $\frac{\partial}{\partial \mathbf{t}} \log \text{softmax}(\mathbf{t})_k = (\bar{\delta}_k - \text{softmax}(\mathbf{t}))$
- $\frac{\partial H_{w_{i+1}}}{\partial U} = -\frac{\partial}{\partial U} \log \text{softmax}(U\mathbf{h}_i)_{w_{i+1}} = (\bar{\delta}_{w_{i+1}} - \text{softmax}(U\mathbf{h}_i)) \otimes \mathbf{h}_i$
- $\frac{\partial H_{w_{i+1}}}{\partial W} = -\frac{\partial}{\partial W} \log \text{softmax}(U\mathbf{h}_i)_{w_{i+1}}$

- Нека положим $\mathbf{z}_i = W\mathbf{h}_{i-1} + VE\chi_{w_i}$. Тогава $\mathbf{h}_i = g(\mathbf{z}_i)$.
- Означения:
  - $g'(\mathbf{a})$ е диагонална матрица с диагонал $g'(\mathbf{a}_i)$.
  - Ако $A \in \mathbb{R}^{L \times M}$е матрица и $\mathbf{b} \in \mathbb{R}^N$ е вектор то $A \otimes \mathbf{b} \in \mathbb{R}^{L \times M \times N}$ и $(A \otimes \mathbf{b})_{k,i,j} = A_{k,i}\mathbf{b}_j$.
  - $\mathbf{I}_N \in \mathbb{R}^{N \times N}$ е единичната матрица.

- $$\frac{\partial H_{w_{i+1}}}{\partial W} = \frac{\partial H_{w_{i+1}}}{\partial \mathbf{h}_i}\frac{\partial \mathbf{h}_i}{\partial W} = (\bar{\delta}_{w_{i+1}} - \text{softmax}(U\mathbf{h}_i))^\top U\frac{\partial \mathbf{h}_i}{\partial W}$$
- $$\frac{\partial \mathbf{h}_i}{\partial W} = \frac{\partial}{\partial W}g(W\mathbf{h}_{i-1} + VE\chi_{w_i}) = g'(\mathbf{z}_i)\left(\mathbf{I}_N \otimes \mathbf{h}_{i-1} + W\frac{\partial \mathbf{h}_{i-1}}{\partial W}\right)$$
- $$\frac{\partial H_{w_{i+1}}}{\partial V} = \frac{\partial H_{w_{i+1}}}{\partial \mathbf{h}_i}\frac{\partial \mathbf{h}_i}{\partial V} = (\bar{\delta}_{w_{i+1}} - \text{softmax}(U\mathbf{h}_i))^\top U\frac{\partial \mathbf{h}_i}{\partial V}$$
- $$\frac{\partial \mathbf{h}_i}{\partial V} = \frac{\partial}{\partial V}g(W\mathbf{h}_{i-1} + VE\chi_{w_i}) = g'(\mathbf{z}_i)\left(W\frac{\partial \mathbf{h}_{i-1}}{\partial V} + \mathbf{I}_N \otimes E\chi_{w_i}\right)$$
- $$\frac{\partial H_{w_{i+1}}}{\partial E} = \frac{\partial H_{w_{i+1}}}{\partial \mathbf{h}_i}\frac{\partial \mathbf{h}_i}{\partial E} = (\bar{\delta}_{w_{i+1}} - \text{softmax}(U\mathbf{h}_i))^\top U\frac{\partial \mathbf{h}_i}{\partial E}$$
- $$\frac{\partial \mathbf{h}_i}{\partial E} = \frac{\partial}{\partial E}g(W\mathbf{h}_{i-1} + VE\chi_{w_i}) = g'(\mathbf{z}_i)\left(W\frac{\partial \mathbf{h}_{i-1}}{\partial E} + V \otimes \chi_{w_i}\right)$$

- $$\frac{\partial H_{w_{i+1}}}{\partial W} = (\bar{\delta}_{w_{i+1}} - \text{softmax}(U\mathbf{h}_i))^\top U g'(\mathbf{z}_i) \sum_{j=1}^{i} \left(\prod_{k=1}^{j-1} W g'(\mathbf{z}_{i-k}) \right) \mathbf{I}_N \otimes \mathbf{h}_{i-j}$$
- $$\frac{\partial H_{w_{i+1}}}{\partial V} = (\bar{\delta}_{w_{i+1}} - \text{softmax}(U\mathbf{h}_i))^\top U g'(\mathbf{z}_i) \sum_{j=1}^{i} \left(\prod_{k=1}^{j-1} W g'(\mathbf{z}_{i-k}) \right) \mathbf{I}_N \otimes E\chi_{i-j+1}$$
- $$\frac{\partial H_{w_{i+1}}}{\partial E} = (\bar{\delta}_{w_{i+1}} - \text{softmax}(U\mathbf{h}_i))^\top U g'(\mathbf{z}_i) \sum_{j=1}^{i} \left(\prod_{k=1}^{j-1} W g'(\mathbf{z}_{i-k}) \right) V \otimes \chi_{i-j+1}$$
- Разглеждаме $\prod_{k=1}^{j-1} W g'(\mathbf{z}_{i-k})$ — съответства на градиента $\frac{\partial \mathbf{h}_i}{\partial \mathbf{h}_{i-j}}$.
- Как зависи нормата на градиента от разстоянието за пропагиране i ?

Операторна норма на матрица

- Нека $A \in \mathbb{R}^{M \times N}$ е матрица. **Операторната норма** на A дефинираме като

$$\|A\| = \sup_{\mathbf{x} \in \mathbb{R}^N} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}$$

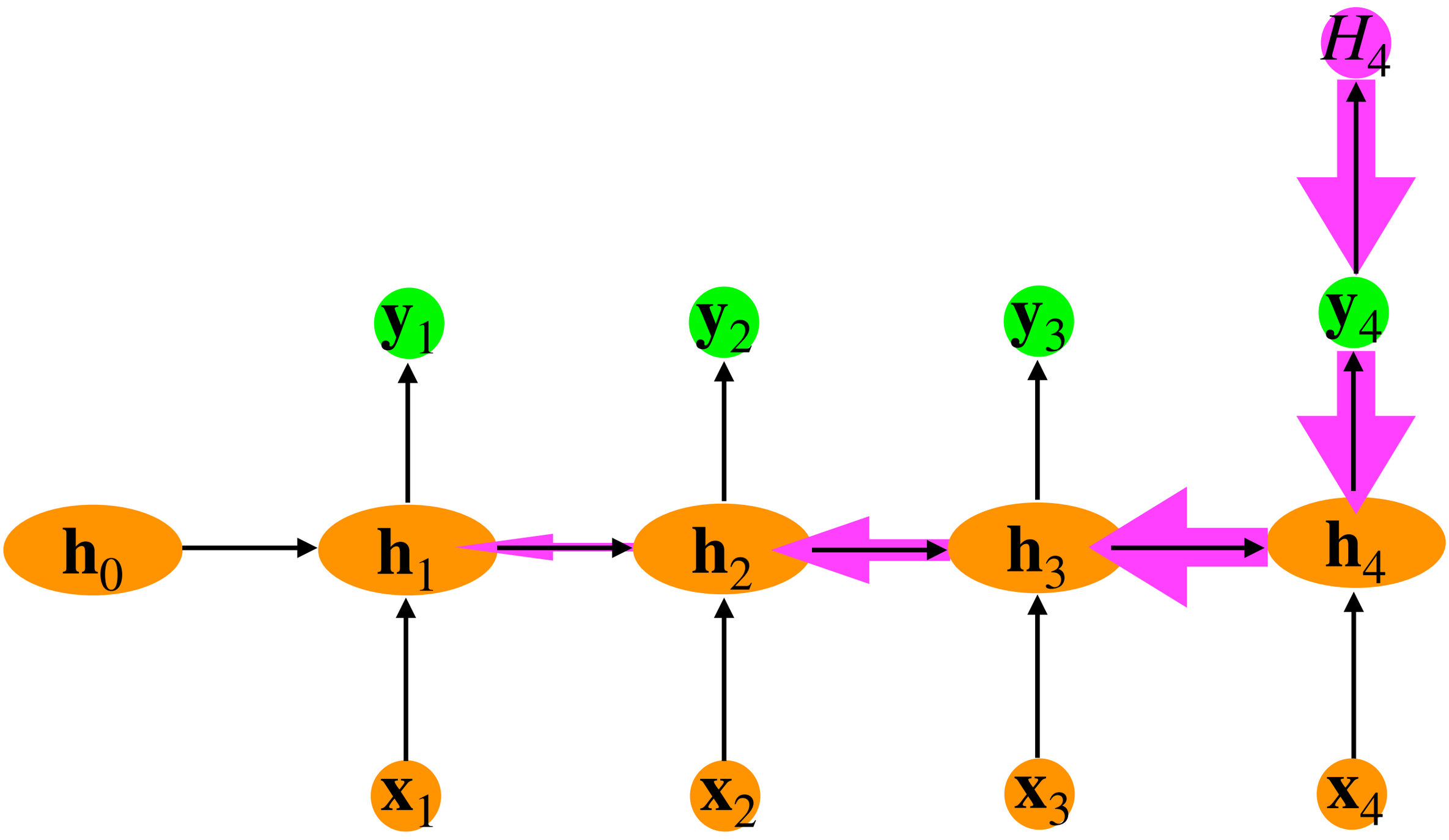
- Свойства:
  - $\|A\mathbf{x}\| \leq \|A\|\|\mathbf{x}\|$
  - $\|AB\| \leq \|A\|\|B\|$, за всеки $A \in \mathbb{R}^{M \times N}$ и $B \in \mathbb{R}^{N \times K}$
  - $\|A\| = |\lambda|$, където λ е най-голямата по модул собствена стойност на A.

- Ако функцията $g(z) = \sigma(z) = \dfrac{1}{1 + e^{-z}}$ то $\|g'(\mathbf{z})\| \leq 1/4$.

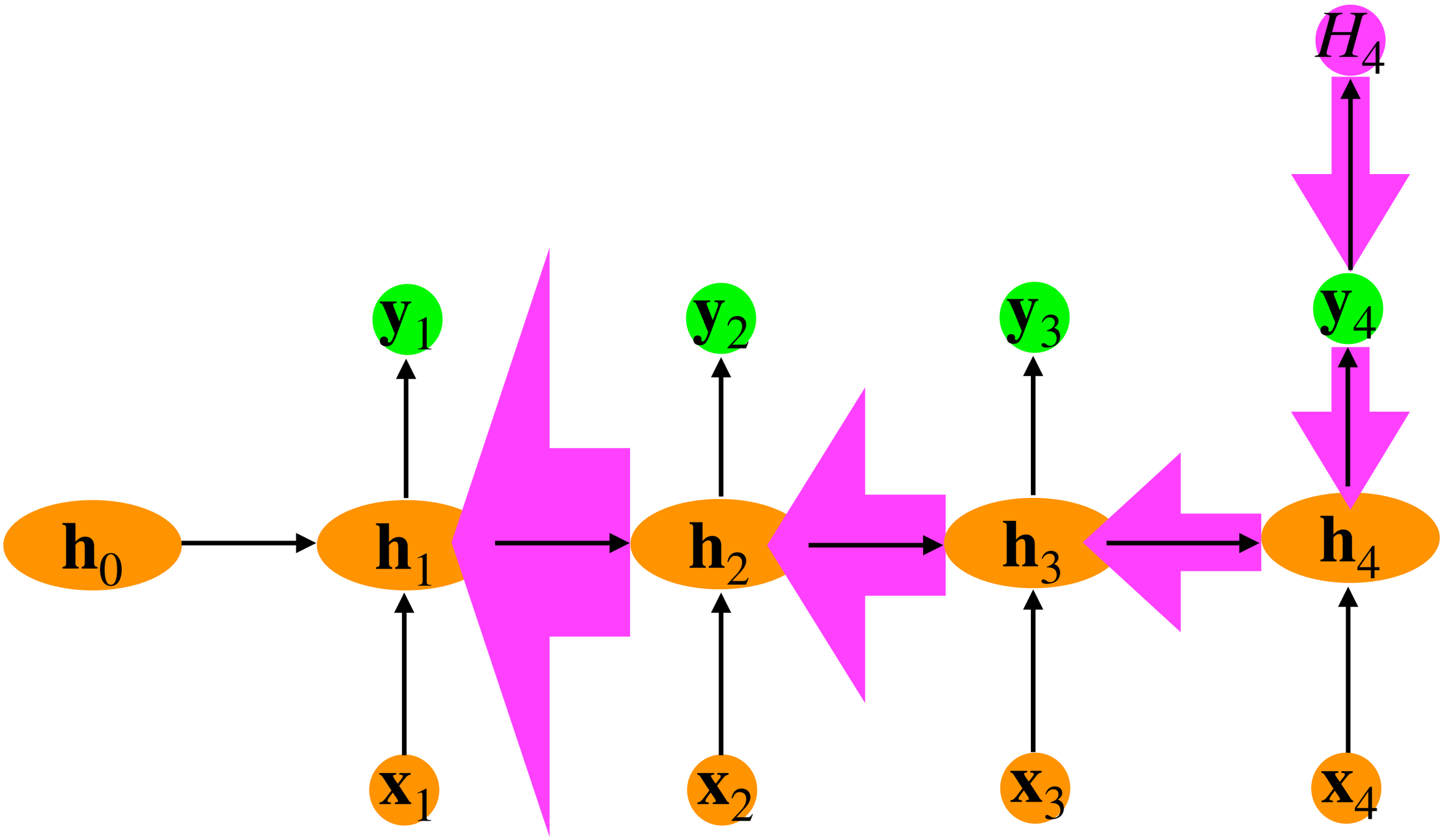
- Ако $\|W\| < 4$ то $\|Wg'(\mathbf{z}_{i-k})\| < 1$. Следователно градиента $\dfrac{\partial \mathbf{h}_i}{\partial \mathbf{h}_{i-j}} = \prod_{k=1}^{j-1} Wg'(\mathbf{z}_{i-k})$ намалява експоненциално — **ИЗЧЕЗВАЩ ГРАДИЕНТ**

- Ако $\|W\| > 4$ то градиента $\dfrac{\partial \mathbf{h}_i}{\partial \mathbf{h}_{i-j}} = \prod_{k=1}^{j-1} Wg'(\mathbf{z}_{i-k})$ евентуално може да расте експоненциално — **ЕКСПЛОДИРАЩ ГРАДИЕНТ**

Пропагиране при рекурентни невронни мрежи



Пропагиране при рекурентни невронни мрежи



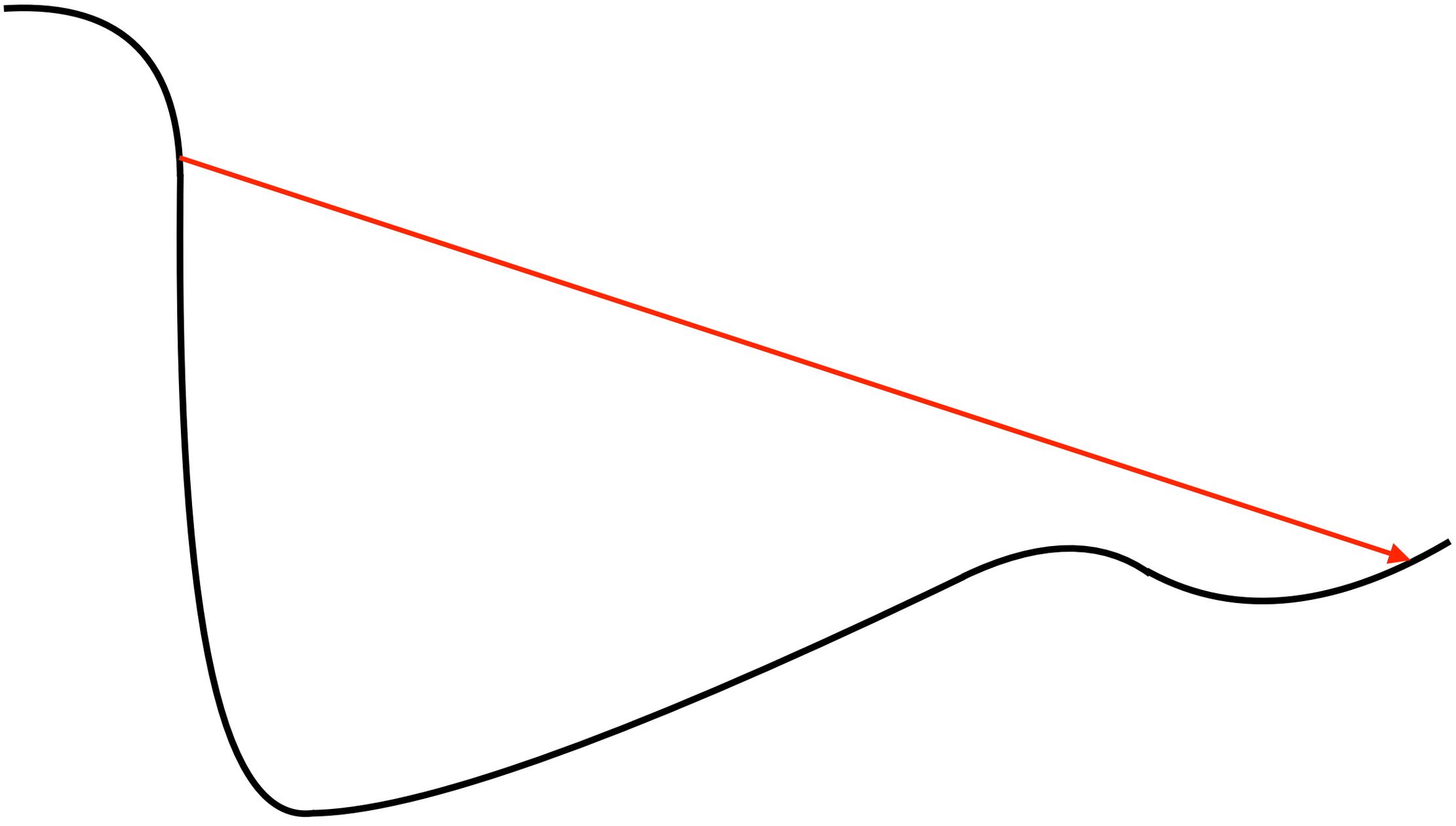
План на лекцията

1. Формалности за курса (3 мин)
2. Особености при обучение на рекурентна невронна мрежа (30 мин)
3. **Проблем и решение при експлодиращ градиент (10 мин)**
4. Проблем при изчезващ градиент (10 мин)
5. Архитектури за решаване на проблема изчезващ градиент (20 мин)
6. Двупосочни и многослойни архитектури с рекурентни невронни мрежи (10 мин)
7. Приложения на рекурентните невронни мрежи за езиков модел и класификация на документи (10 мин)

Проблеми при експлодиращ градиент

- При спускане по градиента имаме:
  - $\theta_{k+1} = \theta_k - \alpha \nabla_\theta H(\theta_k)$
- Има опасност градиента да излезе извън обхвата на числата с плаваща запетая и да получим стойност **`Inf`** или **`NaN`**
- Ако градиента е много голям ще направим огромен скок при спускането по градиента

Проблеми при екплодиращ градиент



Решение: ограничаване на градиента — gradient clipping

- Ако нормата на градиента е над даден праг $\kappa > 0$, то преди да направим спускането намаляваме дължината на градиента до κ.

- По този начин ще направим спускане в същата посока но с по-малка стъпка:

- $\theta_{k+1} = \theta_k - \alpha \ \mathrm{clip}_\kappa(\nabla_\theta H(\theta_k))$, където

$$\mathrm{clip}_\kappa(\mathbf{u}) = \begin{cases} \frac{\kappa}{\|\mathbf{u}\|}\mathbf{u} & \text{if } \|\mathbf{u}\| > \kappa \\ \mathbf{u} & \text{if } \|\mathbf{u}\| \leq \kappa \end{cases}$$

- Решението е просто и се прилага масово в дълбокото машинно обучение при всички невронни архитектури.

План на лекцията

1. Формалности за курса (3 мин)
2. Особености при обучение на рекурентна невронна мрежа (30 мин)
3. Проблем и решение при експлодиращ градиент (10 мин)
4. **Проблем при изчезващ градиент (10 мин)**
5. Архитектури за решаване на проблема изчезващ градиент (20 мин)
6. Двупосочни и многослойни архитектури с рекурентни невронни мрежи (10 мин)
7. Приложения на рекурентните невронни мрежи за езиков модел и класификация на документи (10 мин)

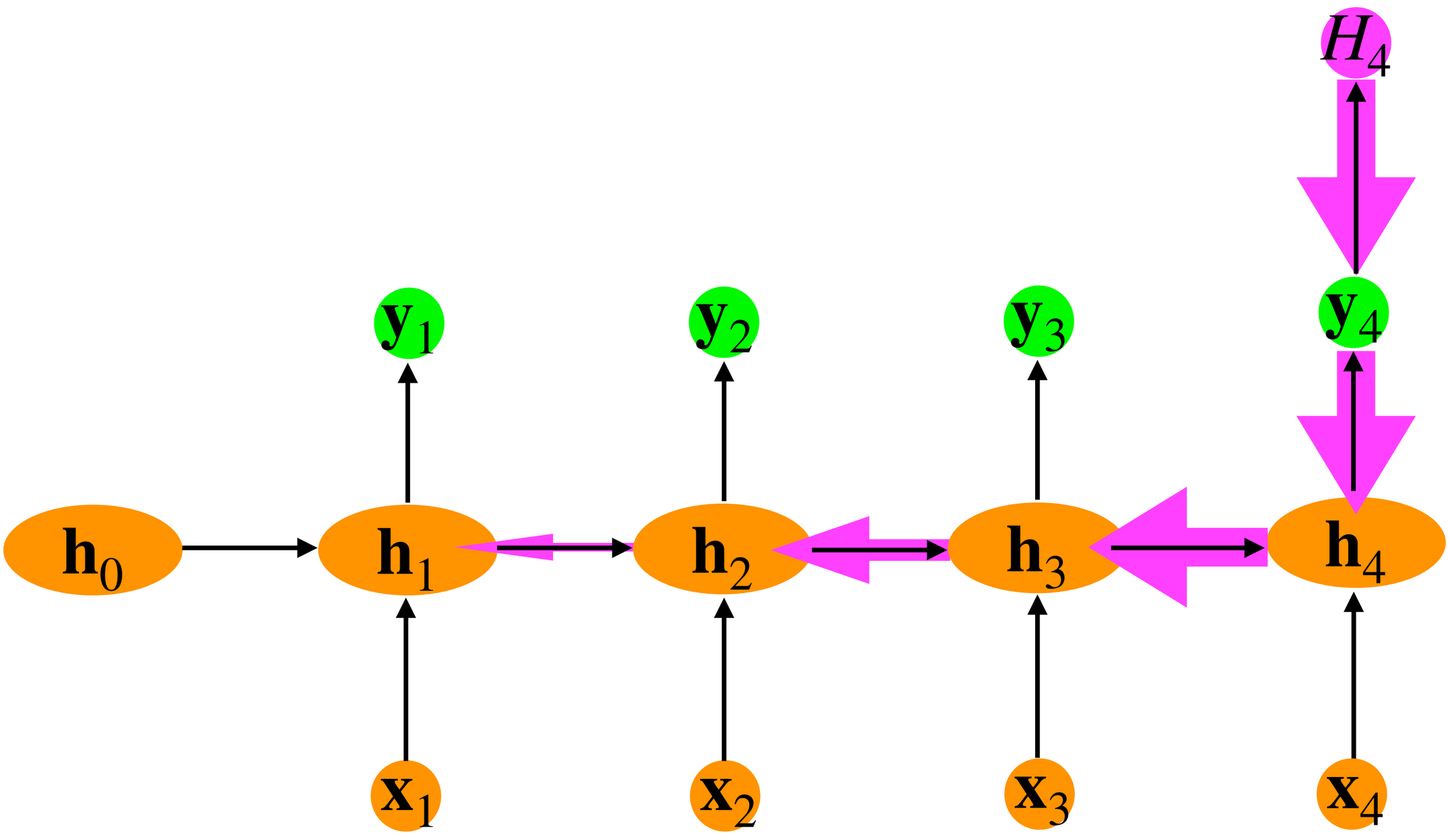
Научаване на зависимости на дълго разстояние

- Пример:
  Когато Иван се опита да отпечата <u>доклада</u> си, той забеляза, че тонерът на принтера е свършил. Той отиде да купи нов тонер от офис магазина. Тонерът беше на разпродажба. След като инсталира новия тонер, Иван най-после успя да отпечата _____.

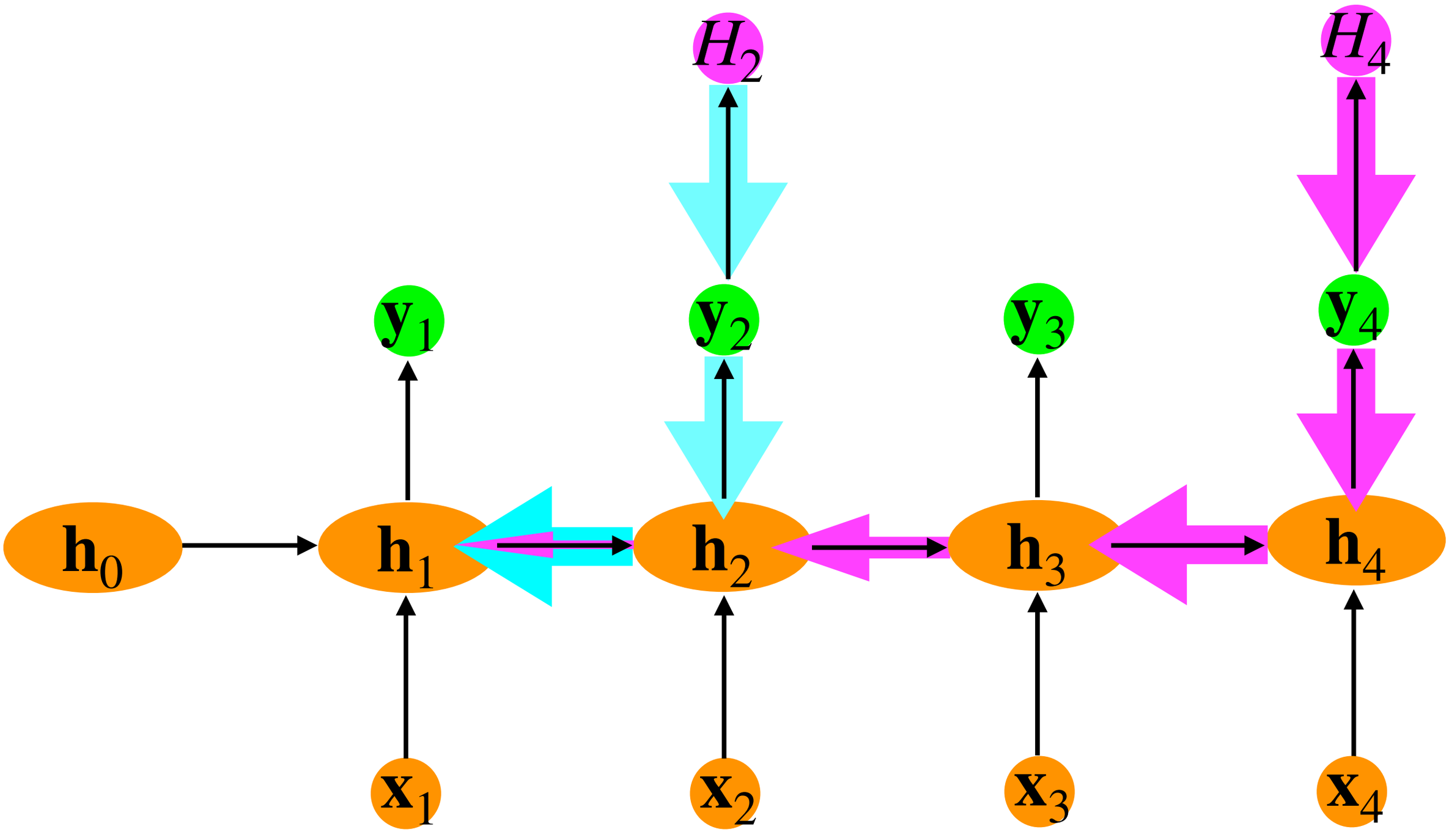
- Моделът е желателно да научи зависимостта между *<u>доклада</u>* на позиция 7 и търсената дума ~32 позиции по-нататък.

- Ако градиента през тези 32 позиции изчезне, моделът няма да може да научи тази зависимост.

Пропагиране при рекурентни невронни мрежи



Пропагиране при рекурентни невронни мрежи близко и далечно разстояние — short term long term

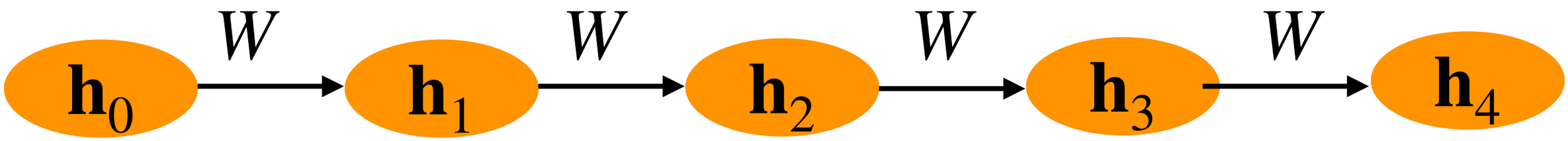


План на лекцията

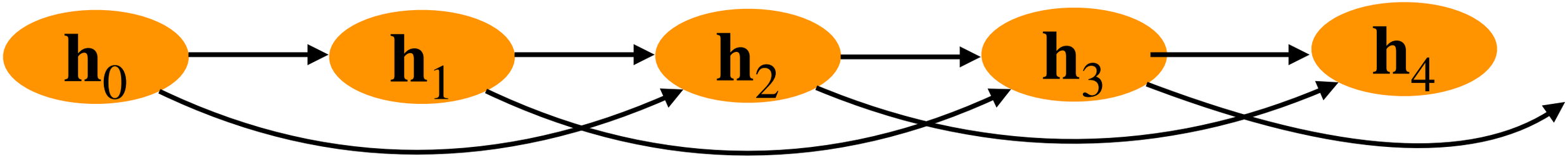
1. Формалности за курса (3 мин)
2. Особености при обучение на рекурентна невронна мрежа (30 мин)
3. Проблем и решение при експлодиращ градиент (10 мин)
4. Проблем при изчезващ градиент (10 мин)
5. **Архитектури за решаване на проблема изчезващ градиент (20 мин)**
6. Двупосочни и многослойни архитектури с рекурентни невронни мрежи (10 мин)
7. Приложения на рекурентните невронни мрежи за езиков модел и класификация на документи (10 мин)

Решаване на проблема с изчезващия градиент

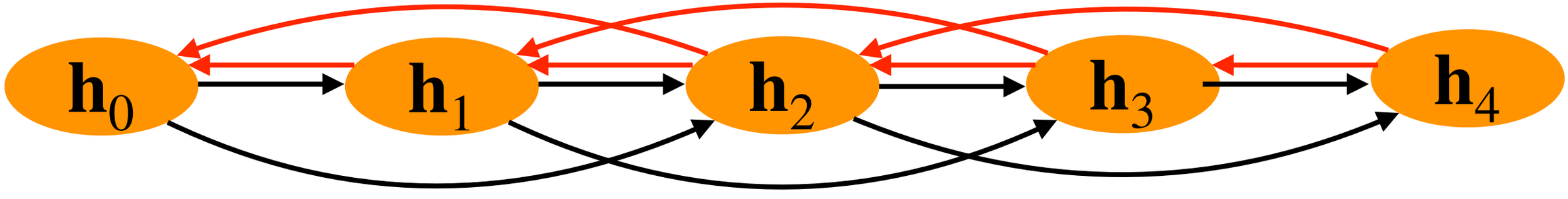
- Рекурентната формула $\mathbf{h}_i = g(W\mathbf{h}_{i-1} + V\mathbf{x}_i)$ води до вдигане на степен на матрицата W при пропагирането на градиента.



- Този проблем може да се реши, ако връзките между отделните състояния станат по-директни.



Решаване на проблема с изчезващия градиент

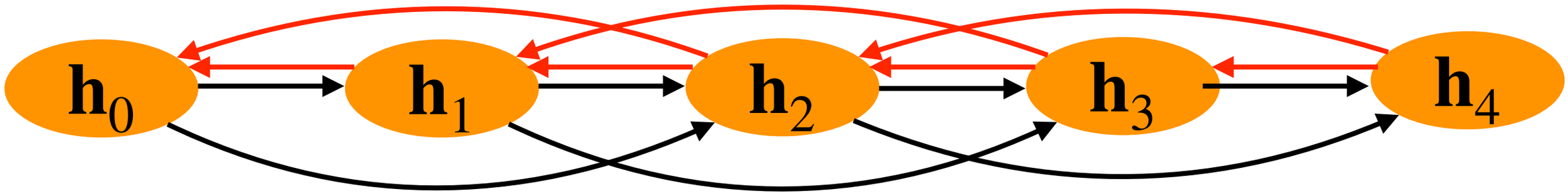


- Такива връзки наричаме **преки връзки** (shortcut connection, skip connection). Подобни методи се използват в много от архитектурите с дълбоки невронни мрежи — skip-net, highway net, …

- През пряката връзка позволяваме на градиента да пропагира директно до предходните състояния.

- Но за да се извърши обучението е необходимо да се контролира информацията по преките връзки.

Контрол на информацията с порти



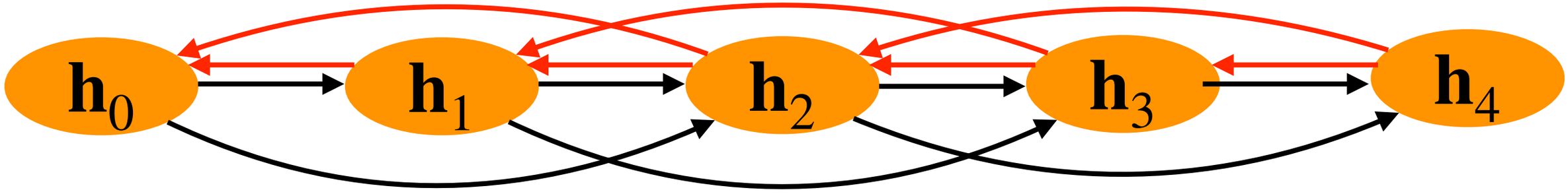
- Нека приложим *адаптивни* преки връзки.

- $h_t = f(h_{t-1}, x_t) = u_t \odot \tilde{h}_t + (1 - u_t) \odot h_{t-1}$

  - кандидат за презапис: $\tilde{h}_t = \tanh(W[h_{t-1}; x_t] + b)$

  - порта за презапис: $u_t = \sigma(W_u[h_{t-1}; x_t] + b_u)$

Контрол на информацията с порти



- Нека позволим да премахнем *адаптивно* ненужни връзки.

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

  - кандидат за презапис: $\tilde{h}_t = \tanh(W[(r_t \odot h_{t-1}); x_t] + b)$

  - порта за презапис: $u_t = \sigma(W_u[h_{t-1}; x_t] + b_u)$

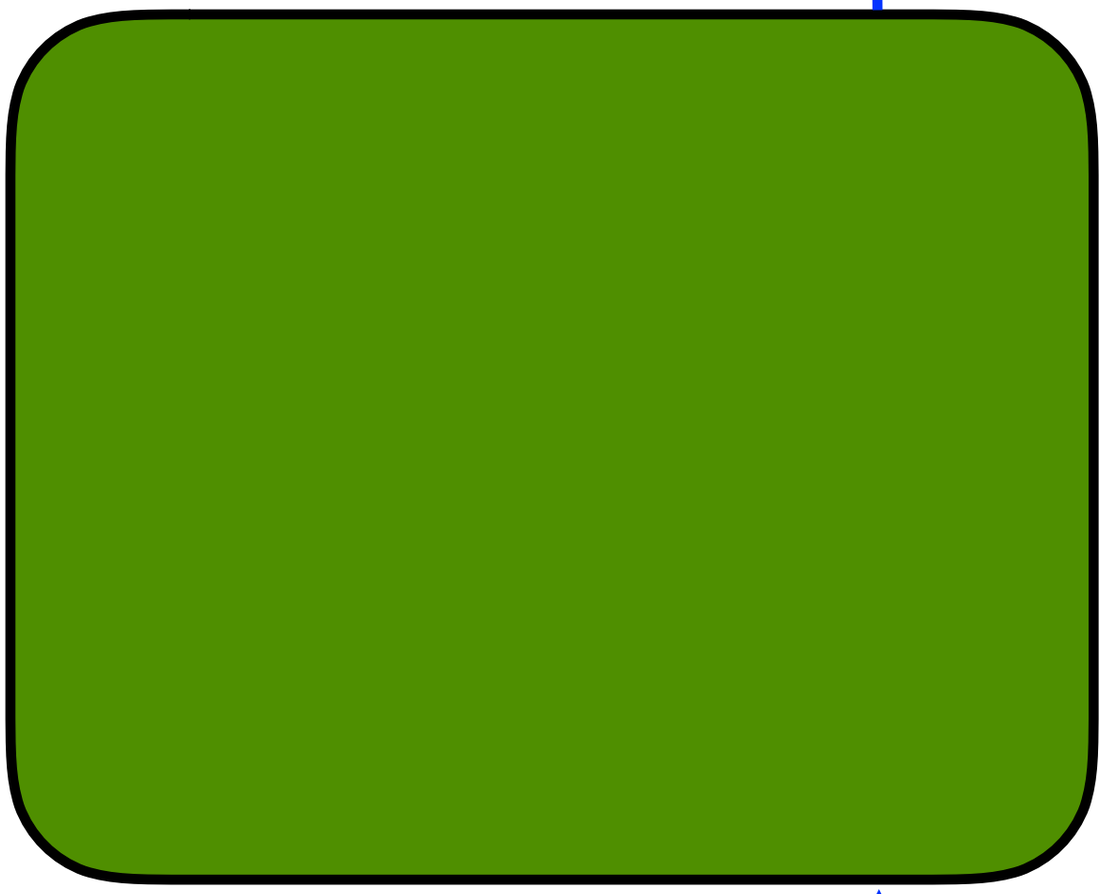
  - порта за нулиране: $r_t = \sigma(W_r[h_{t-1}; x_t] + b_r)$

Рекурентен елемент с порти
Gated recurrent unit GRU

- $h_t = u_t \odot \tilde{h}_t + (1 - u_t) \odot h_{t-1}$
- $\tilde{h}_t = \tanh(W[(r \odot h_{t-1}); x_t] + b)$
- $u_t = \sigma(W_u[h_{t-1}; x_t] + b_u)$
- $r_t = \sigma(W_r[h_{t-1}; x_t] + b_r)$
- $x_t \in \mathbb{R}^M, h_t, \tilde{h}_t, h_{t-1}, u_t, r_t \in \mathbb{R}^N$
- $W, W_u, W_r \in \mathbb{R}^{N \times (N+M)}, b, b_u, b_r \in \mathbb{R}^N$

Интуиция за GRU

Памет / регистри h

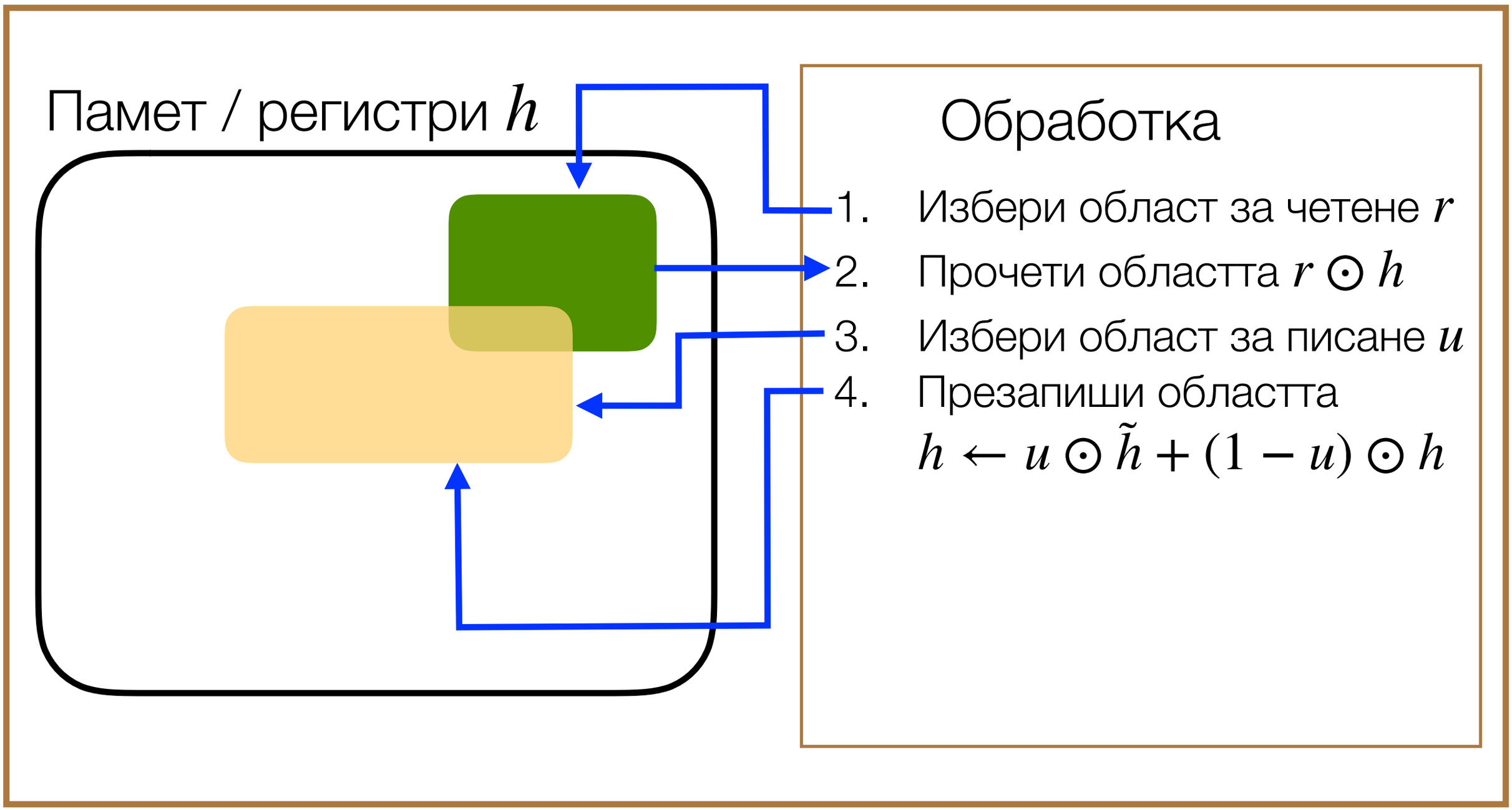


Обработка

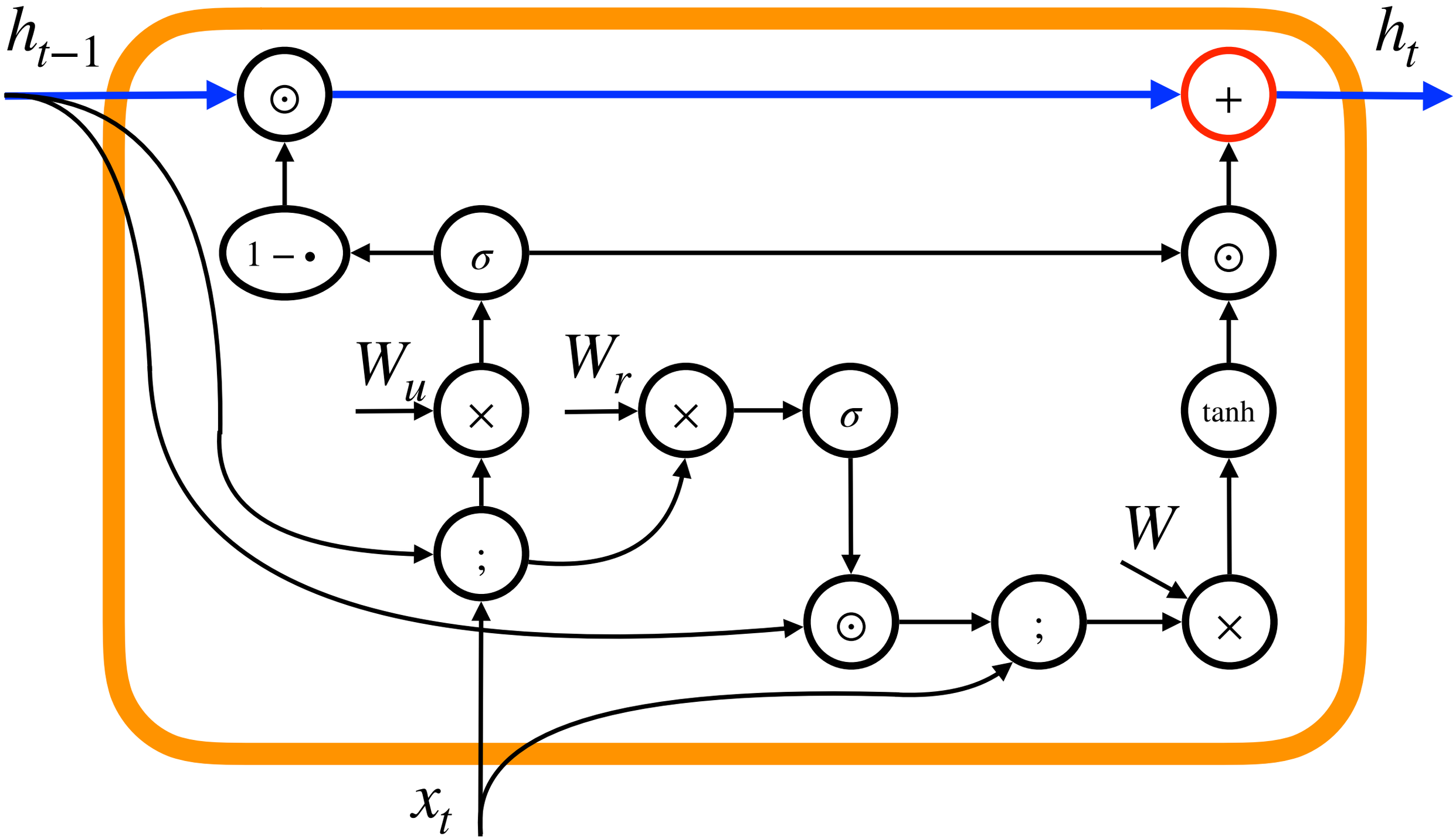
1. Прочети цялата памет h
2. Презапиши цялата памет

$$h \leftarrow g(W[h; x] + b)$$

Интуиция за GRU



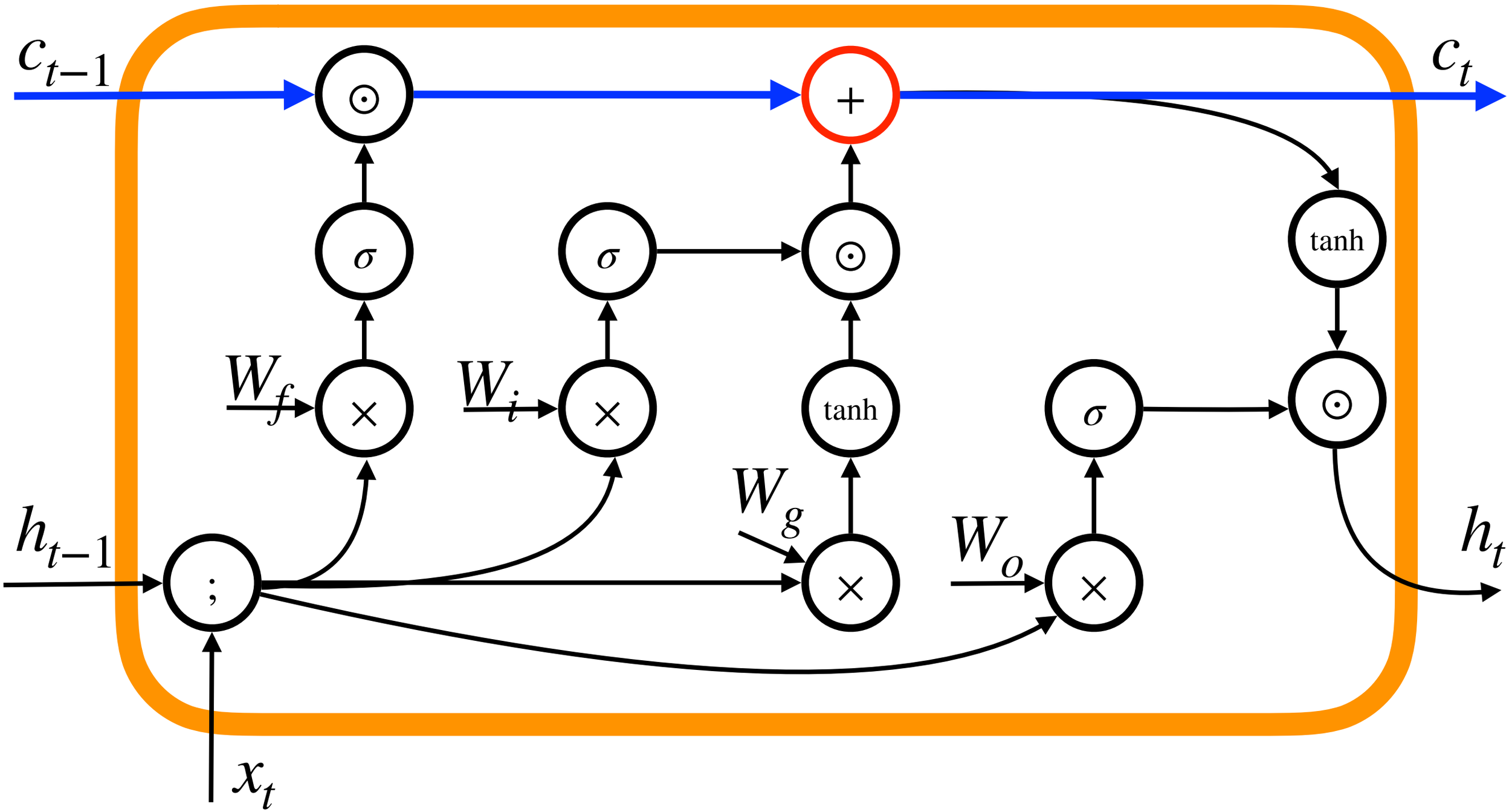
GRU изчислителен граф



Рекурентен елемент с краткосрочна и дългосрочна памет Long-short term memory LSTM

- $h_t = o_t \odot \tanh(c_t)$
- $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$
- $\tilde{c}_t = \tanh(W_c[h_{t-1}; x_t] + b_c)$
- $o_t = \sigma(W_o[h_{t-1}; x_t] + b_o)$
- $i_t = \sigma(W_i[h_{t-1}; x_t] + b_i)$
- $f_t = \sigma(W_f[h_{t-1}; x_t] + b_f)$
- $x_t \in \mathbb{R}^M, h_t, c_t, \tilde{c}_t, c_{t-1}, o_t, i_t, f_t \in \mathbb{R}^N$
- $W_c, W_o, W_i, W_f \in \mathbb{R}^{N \times (N+M)}, b_c, b_o, b_i, b_f \in \mathbb{R}^N$

LSTM изчислителен граф



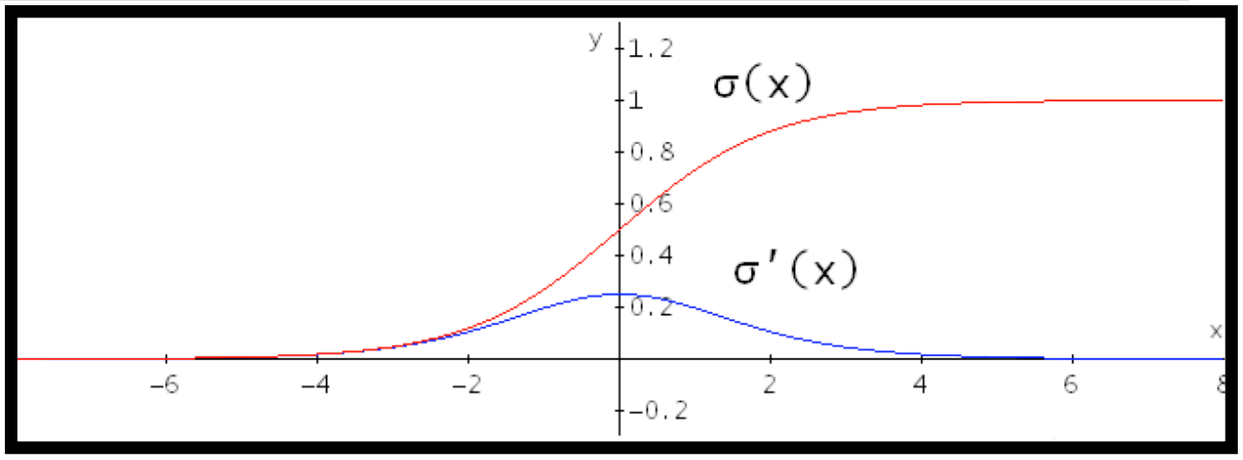
Сравнение между LSTM и GRU

- $h_t = o_t \odot \tanh(c_t)$
- $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$
- $\tilde{c}_t = \tanh(W_c[h_{t-1}; x_t] + b_c)$
- $o_t = \sigma(W_o[h_{t-1}; x_t] + b_o)$
- $i_t = \sigma(W_i[h_{t-1}; x_t] + b_i)$
- $f_t = \sigma(W_f[h_{t-1}; x_t] + b_f)$

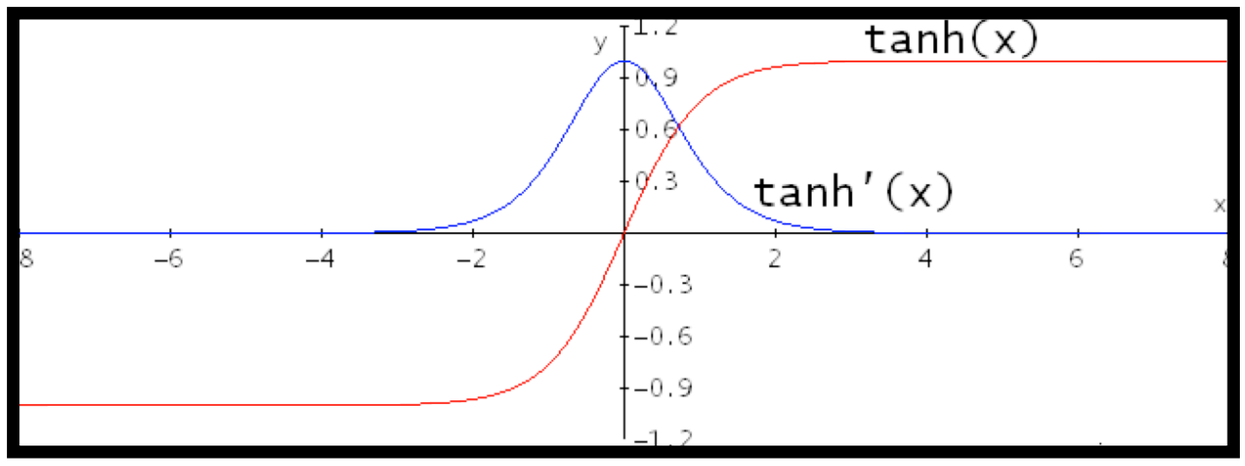
- $h_t = u_t \odot \tilde{h}_t + (1 - u_t) \odot h_{t-1}$
- $\tilde{h}_t = \tanh(W[(r \odot h_{t-1}); x_t] + b)$
- $u_t = \sigma(W_u[h_{t-1}; x_t] + b_u)$
- $r_t = \sigma(W_r[h_{t-1}; x_t] + b_r)$

Нелинейни активационни функции

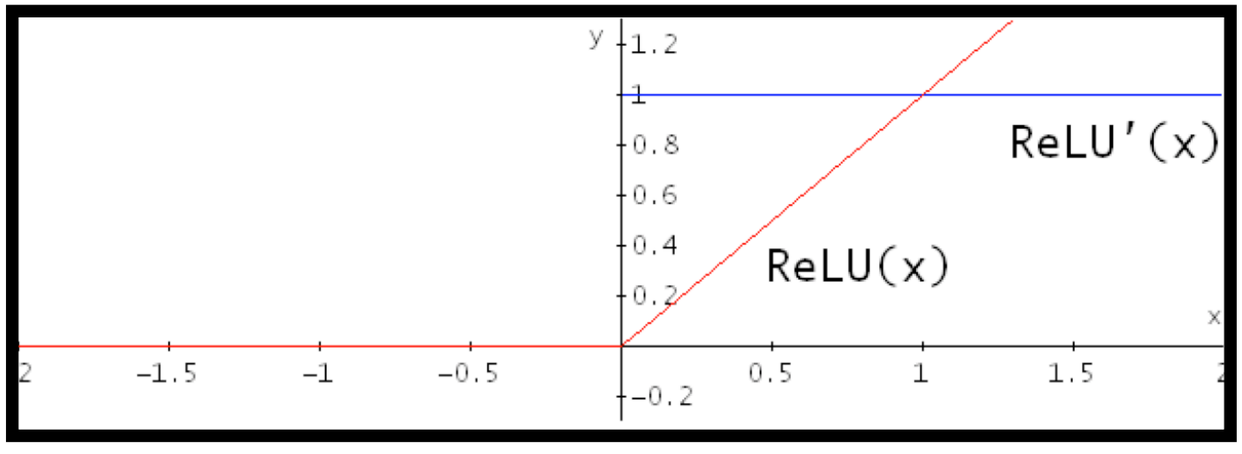
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} = 2\sigma(2x) - 1$$



$$\mathrm{ReLU}(x) = \begin{cases} x & x > 0 \\ 0 & \text{otherwise} \end{cases}$$



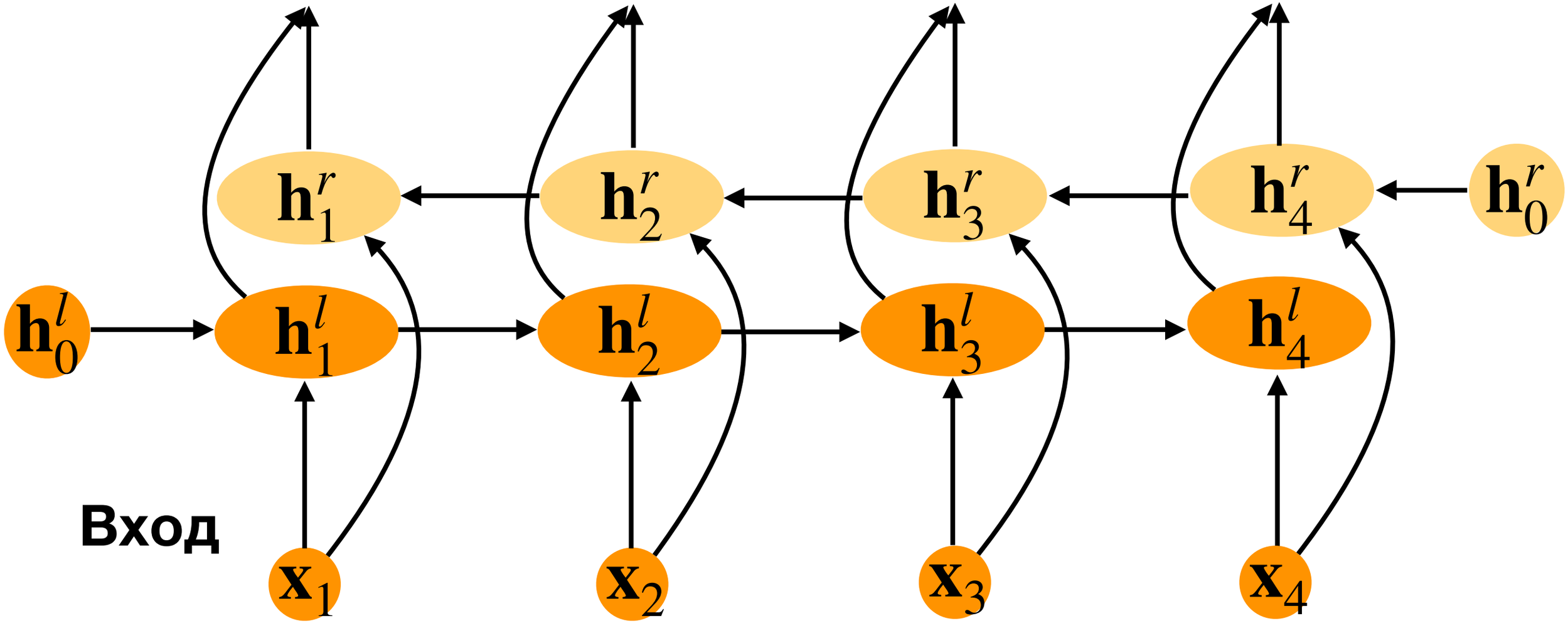
Дискусия за рекурентни мрежи с LSTM / GRU елементи

- Чрез използването на портали при тези архитектури се осъществява ефективно обучение на зависимости на дълго разстояние
- LSTM и GRU са най-широко използваните елементи за рекурентни невронни мрежи с портали
- Всички съвременни платформи за дълбоки невронни мрежи имат готови оптимизирани имплементации на LSTM и GRU елементи
- При използването на партиден стохастичен градиент остават проблемите с различните дължини на последователностите
- Няма еднозначен отговор коя от двете архитектури е по-добра
- Рекурентни архитектури с използване на LSTM и GRU елементи са съставна част на повечето съвременни дълбоки невронни мрежи

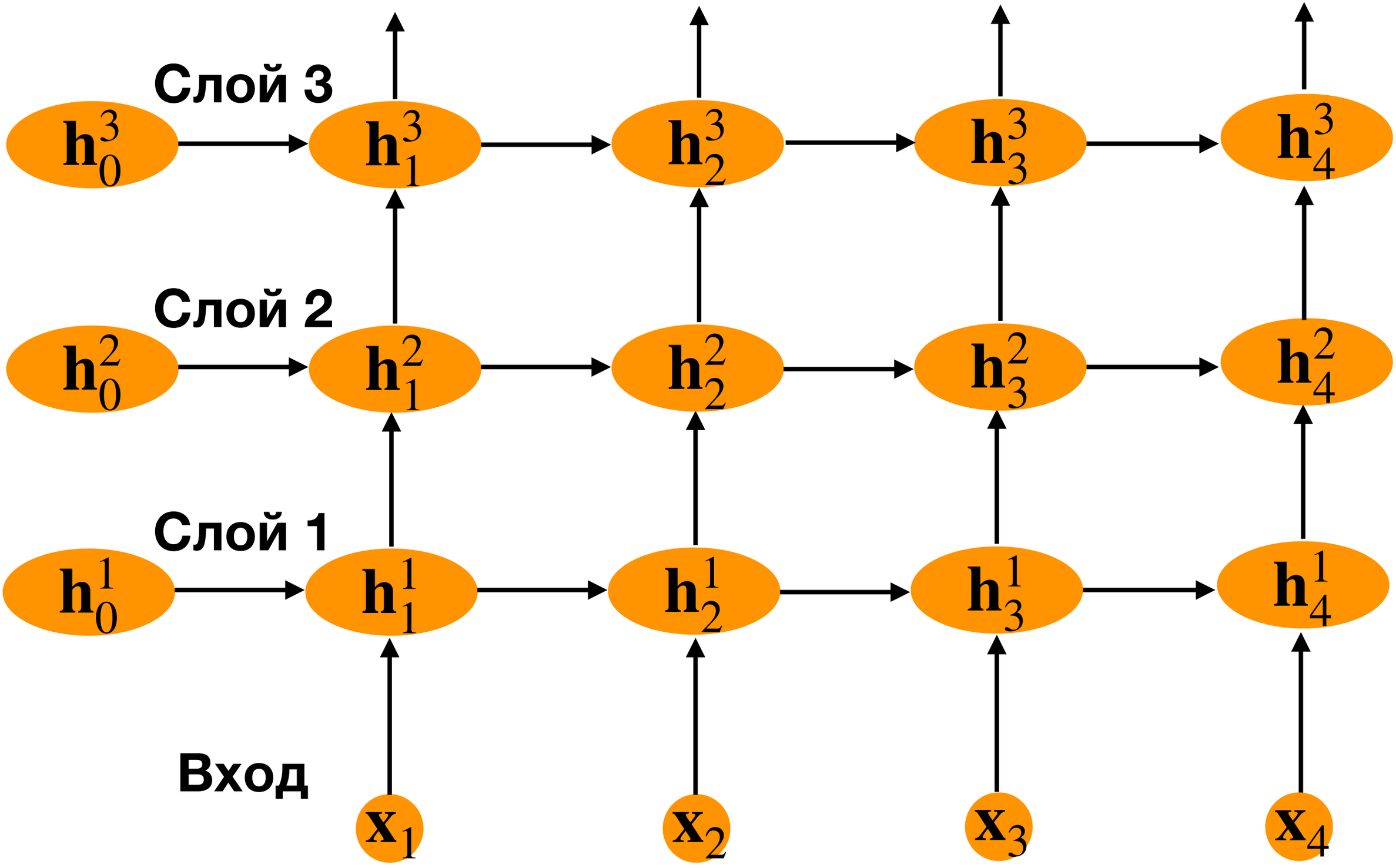
План на лекцията

1. Формалности за курса (3 мин)
2. Особености при обучение на рекурентна невронна мрежа (30 мин)
3. Проблем и решение при експлодиращ градиент (10 мин)
4. Проблем при изчезващ градиент (10 мин)
5. Архитектури за решаване на проблема изчезващ градиент (20 мин)
6. **Двупосочни и многослойни архитектури с рекурентни невронни мрежи (10 мин)**
7. Приложения на рекурентните невронни мрежи за езиков модел и класификация на документи (10 мин)

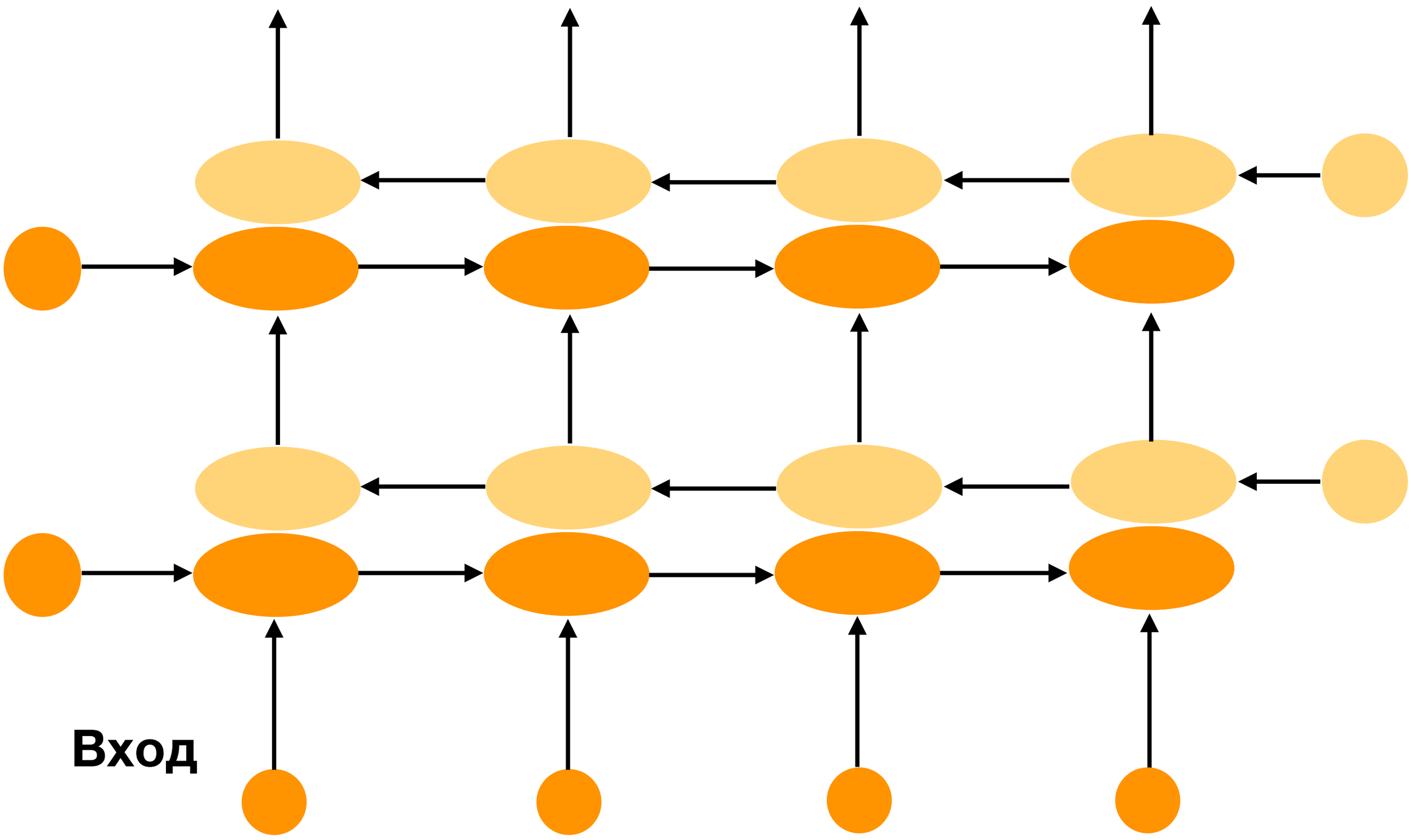
Двупосочни рекурентни невронни мрежи



Многослойни рекурентни невронни мрежи



Двупосочни многослойни рекурентни невронни мрежи



План на лекцията

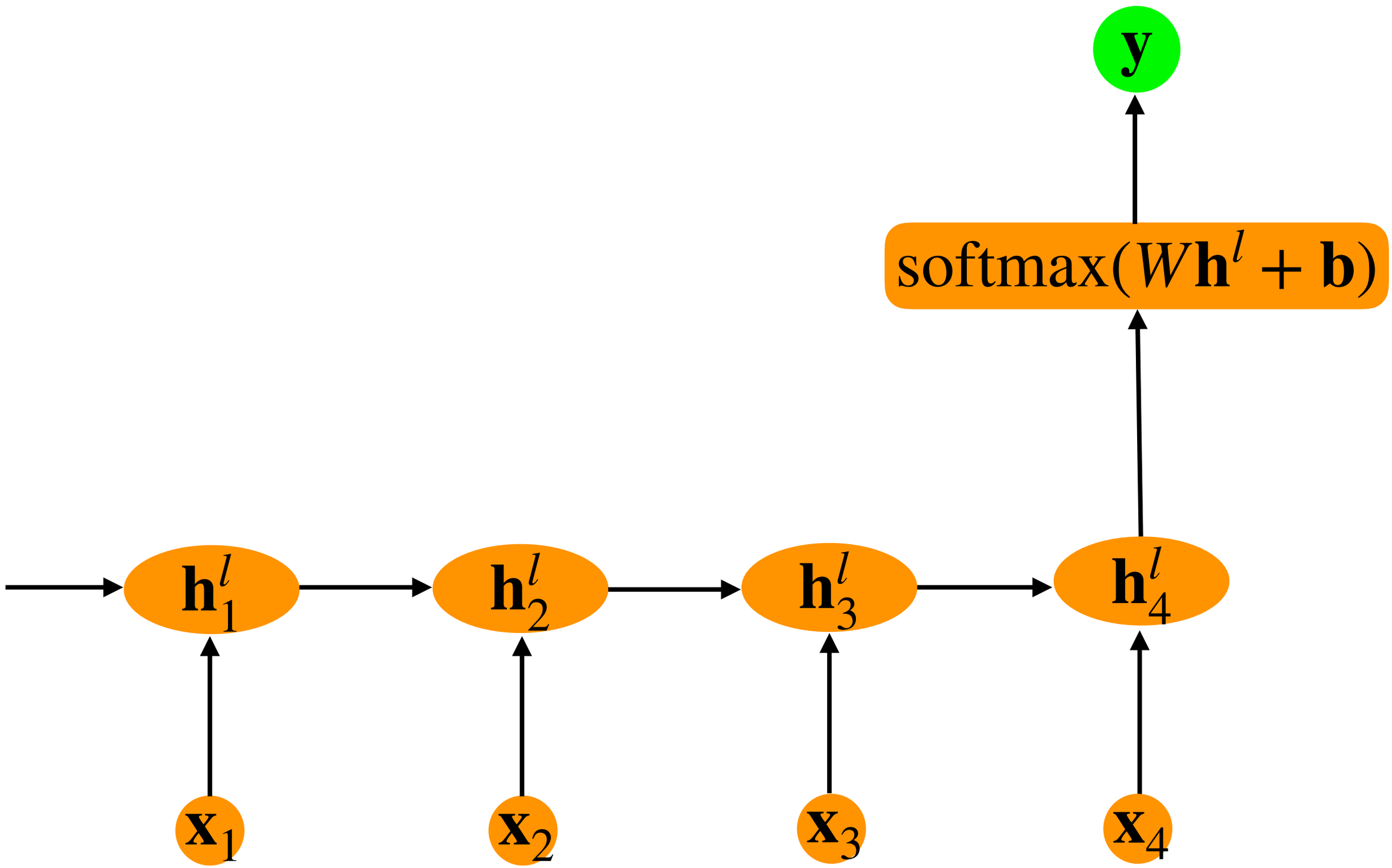
1. Формалности за курса (3 мин)
2. Особености при обучение на рекурентна невронна мрежа (30 мин)
3. Проблем и решение при експлодиращ градиент (10 мин)
4. Проблем при изчезващ градиент (10 мин)
5. Архитектури за решаване на проблема изчезващ градиент (20 мин)
6. Двупосочни и многослойни архитектури с рекурентни невронни мрежи (10 мин)
7. **Приложения на рекурентните невронни мрежи за езиков модел и класификация на документи (10 мин)**

Приложение на РНН за езиков модел

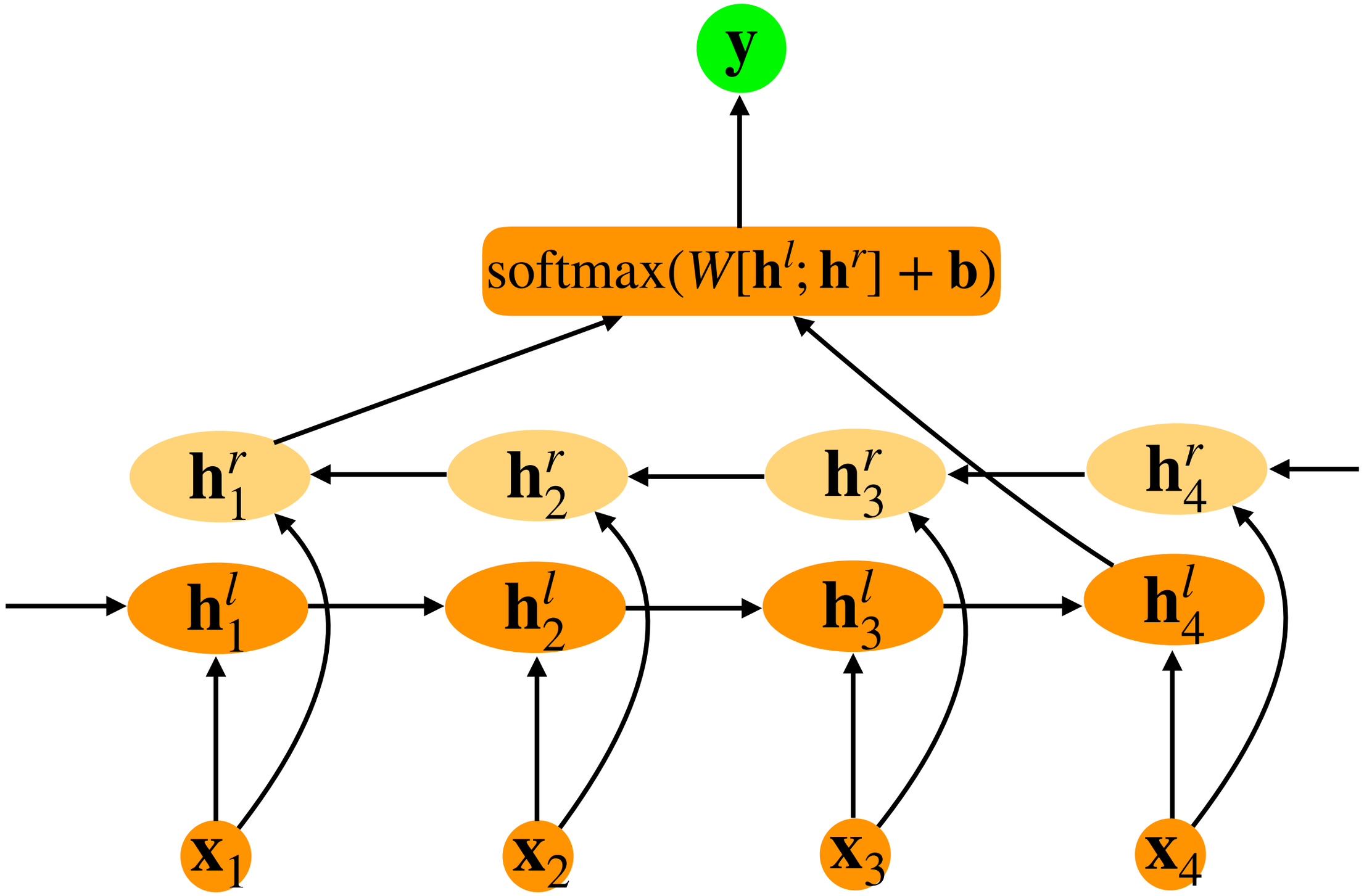
Модел	**Перплексия**
3-грамен с изглаждане	71
Word2Vec CBOW	56
EM на Bengio et al.	39
LSTM RNN	32
LSTM Bi-RNN	**11***

*** Перплексията изчислена при двупосочния модел не е коректна от вероятностна гледна точка и не може директно да се сравнява с перплексията на другите методи.

Приложение на РНН за класификация на документи



Приложение на РНН за класификация на документи



Приложение на РНН класификация на документи

Модел	**F1**
Наивен Бейсов класификатор	89.9
Логистична регресия върху BOW	92.6
LSTM RNN	94.6
LSTM Bi-RNN	**96.7**

Cross Entropy Loss

Softmax

Linear

dropout

h_1^2 h_2^2 h_3^2 h_4^2 h_5^2 h_6^2

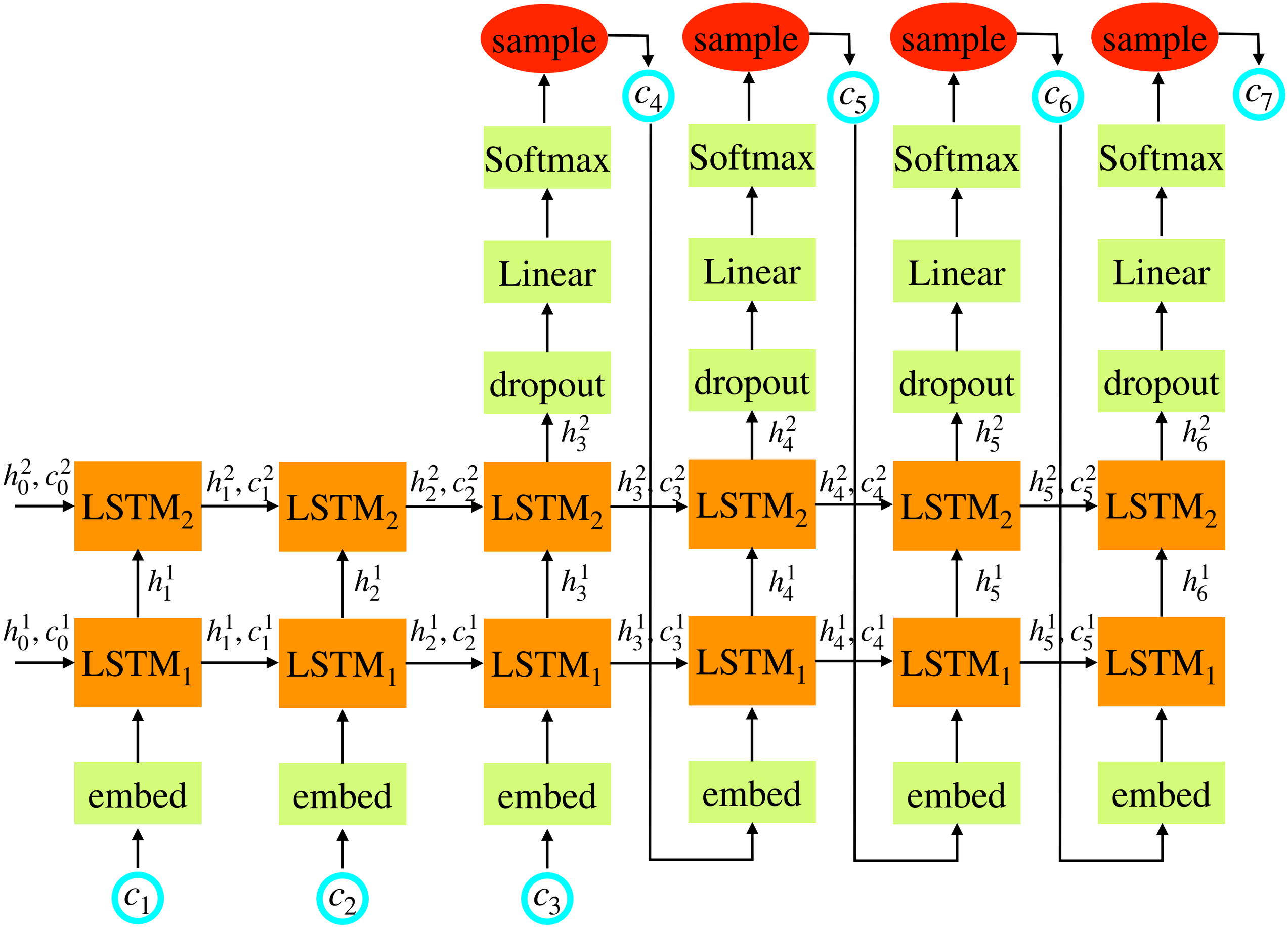
h_0^2, c_0^2 → LSTM_2 → h_1^2, c_1^2 → LSTM_2 → h_2^2, c_2^2 → LSTM_2 → h_3^2, c_3^2 → LSTM_2 → h_4^2, c_4^2 → LSTM_2 → h_5^2, c_5^2 → LSTM_2

h_1^1 h_2^1 h_3^1 h_4^1 h_5^1 h_6^1

h_0^1, c_0^1 → LSTM_1 → h_1^1, c_1^1 → LSTM_1 → h_2^1, c_2^1 → LSTM_1 → h_3^1, c_3^1 → LSTM_1 → h_4^1, c_4^1 → LSTM_1 → h_5^1, c_5^1 → LSTM_1

embed

c_1 c_2 c_3 c_4 c_5 c_6


sample
sample
sample
sample
c_4
c_5
c_6
c_7
Softmax
Softmax
Softmax
Softmax
Linear
Linear
Linear
Linear
dropout
dropout
dropout
dropout
h_3^2
h_4^2
h_5^2
h_6^2
h_0^2, c_0^2
h_1^2, c_1^2
h_2^2, c_2^2
h_3^2, c_3^2
h_4^2, c_4^2
h_5^2, c_5^2
LSTM_2
h_1^1
h_2^1
h_3^1
h_4^1
h_5^1
h_6^1
h_0^1, c_0^1
h_1^1, c_1^1
h_2^1, c_2^1
h_3^1, c_3^1
h_4^1, c_4^1
h_5^1, c_5^1
LSTM_1
embed
c_1
c_2
c_3