Рекуррентттік нейрондық желілер

RNN, LSTM және GRU

Тізбектегі ақпарат қалай сақталады және жаңартылады?

Дәрістің негізгі міндеттері

- Жасырын күй мен уақыт бойынша ортақ параметрлерді түсіндіру.
- ВРТТ және градиенттің өшуі мен өсуі мәселелерін талдау.
- LSTM және GRU қақпаларын, мүмкіндіктері мен шектеулерін салыстыру.

Тізбекті деректер: реттілік неге маңызды?

$$X = (x_1, x_2, \ldots, x_T)$$

Тізбек мысалдары

Сөйлемдегі сөздер.
Аудиосигнал үзінділері.
Температура мен энергия тұтынуы.
Өндірістік датчик өлшемдері.

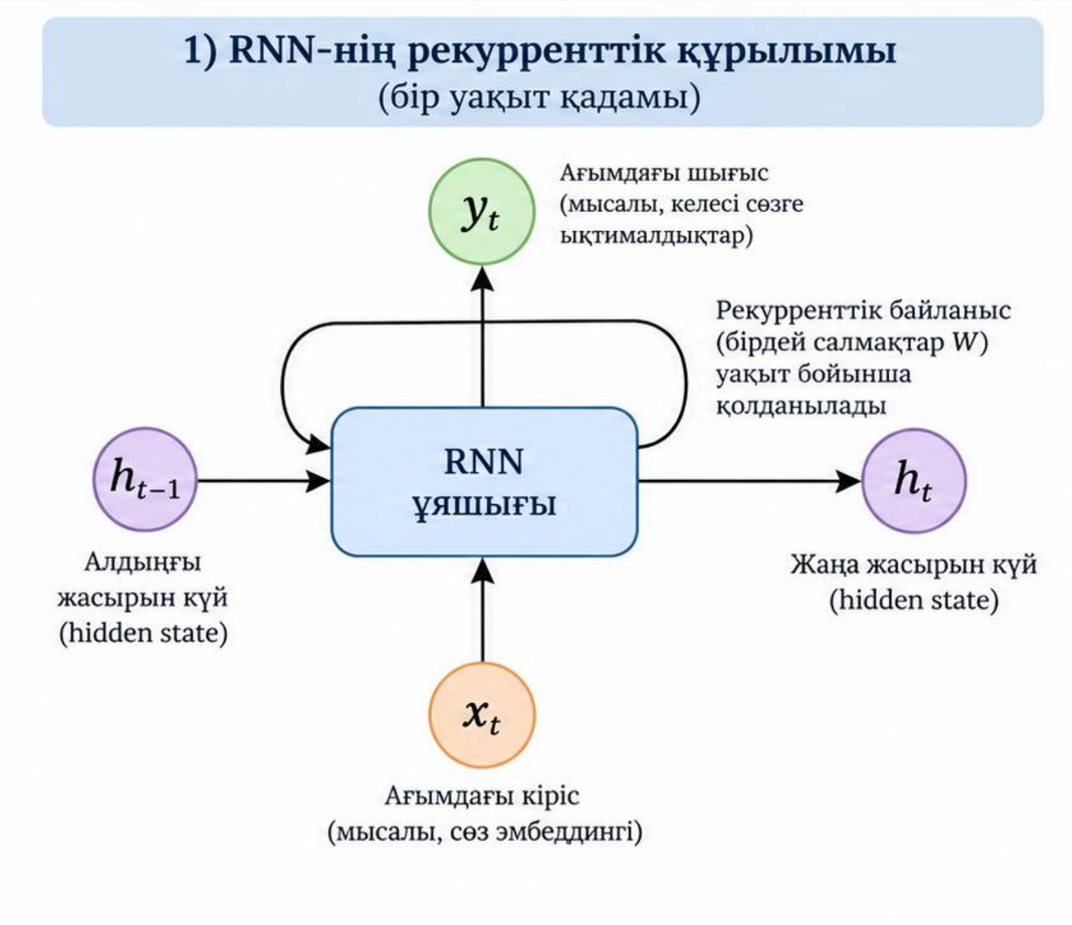
Рекурренттік тәсіл

Ағымдағы кіріспен бірге алдыңғы жасырын күй пайдаланылады.

Реттілік туралы ақпарат күй арқылы тасымалданады.

Жасырын күй — адам жадысының көшірмесі емес, үйренілетін сандық көрініс.

RNN: бір уақыт қадамындағы есептеу



$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

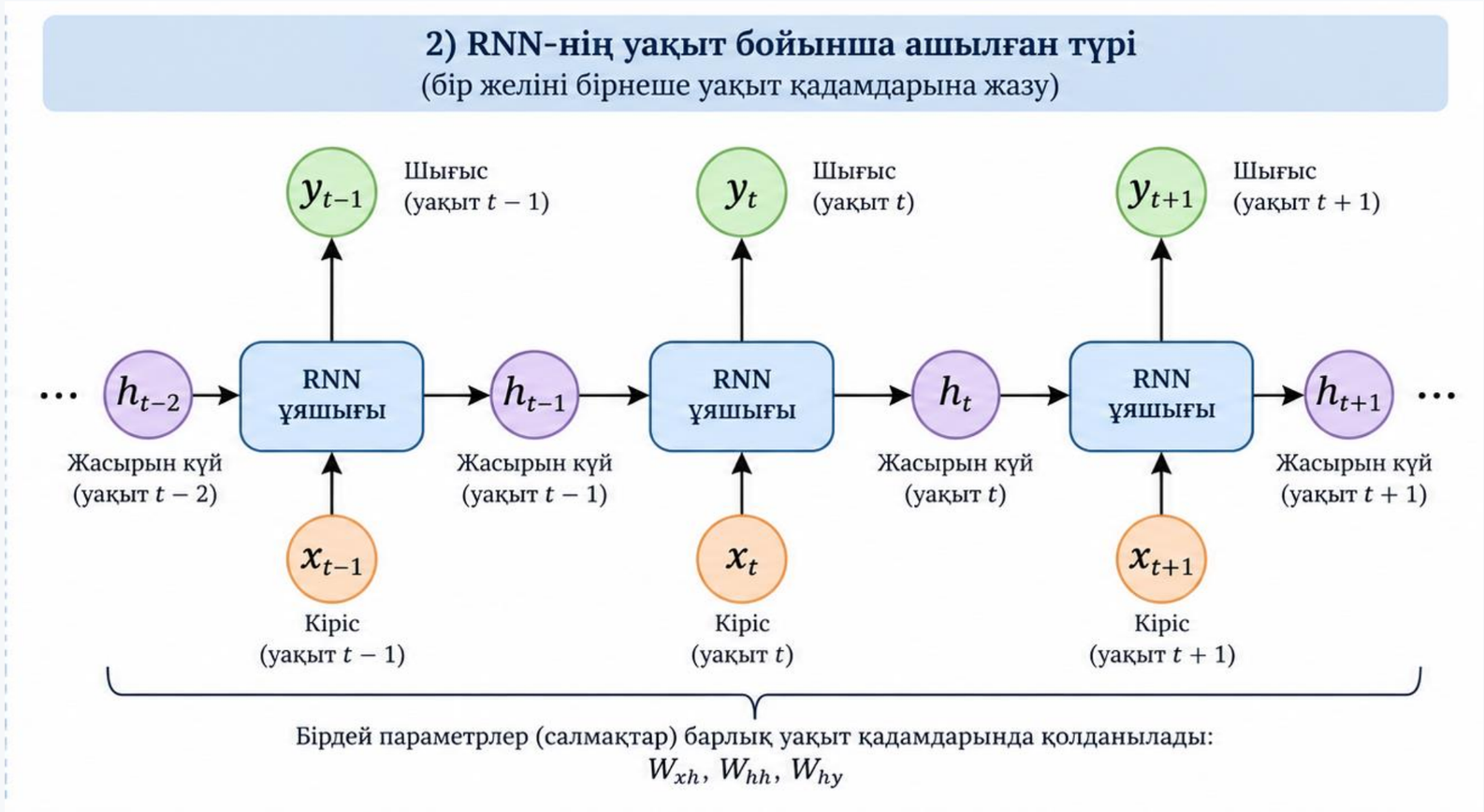
$$\hat{y}_t = g(W_{hy}h_t + b_y)$$

x_t — ағымдағы кіріс
h_{t-1} — алдыңғы жасырын күй
h_t — жаңартылған күй

Қарапайым RNN-де f жиі tanh болады.

Дереккөз: 7-дәріс конспектісі, 2-бет; 7.1-суреттің бірінші бөлігі.

RNN-ді уақыт бойынша ашу



W_{xh}, W_{hh} және W_{hy} барлық уақыт қадамдарында ортақ қолданылады.

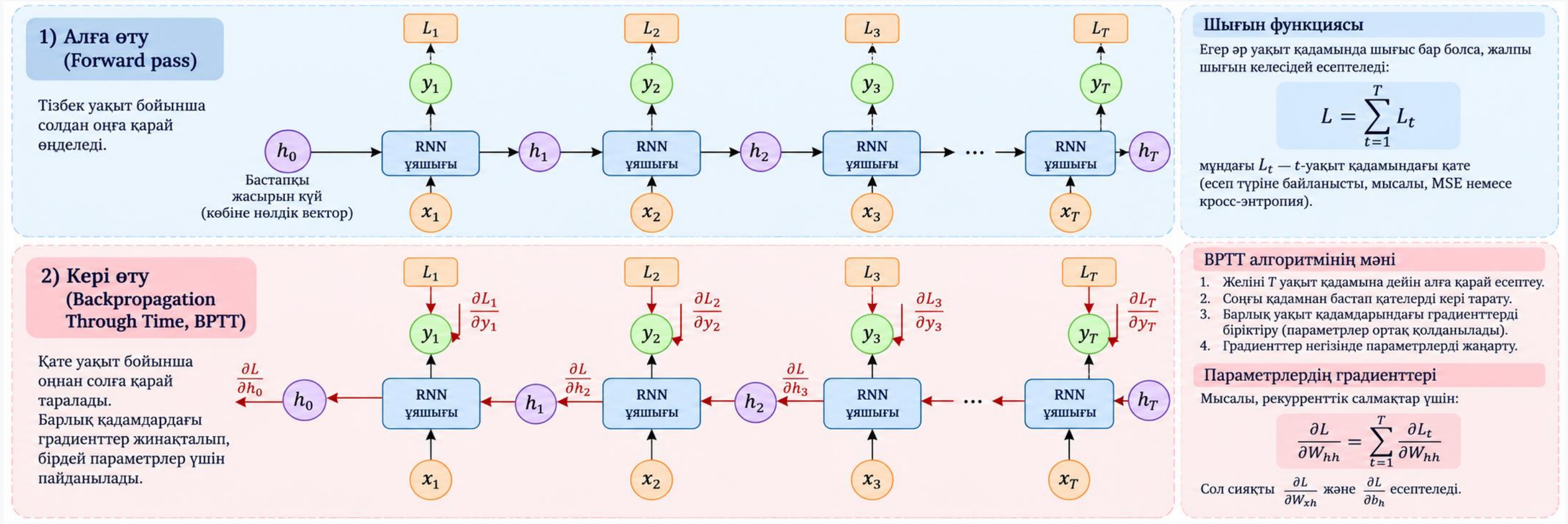
Дереккөз: конспект, 3-бет; 7.1-суреттің екінші бөлігі.

Тізбектің кірісі мен шығысын ұйымдастыру

Ұйымдастыру	Кіріс → шығыс	Мысал
One-to-One	Бір элемент → бір нәтиже	Реттіліксіз базалық есеп
One-to-Many	Бір кіріс → тізбек	Кескінге мәтіндік сипаттама
Many-to-One	Тізбек → бір нәтиже	Мәтін реңкін анықтау
Many-to-Many	Тізбек → тізбек	Әр уақыт қадамында болжау

Шығыс барлық қадамда немесе тек соңғы күй бойынша есептелуі мүмкін.

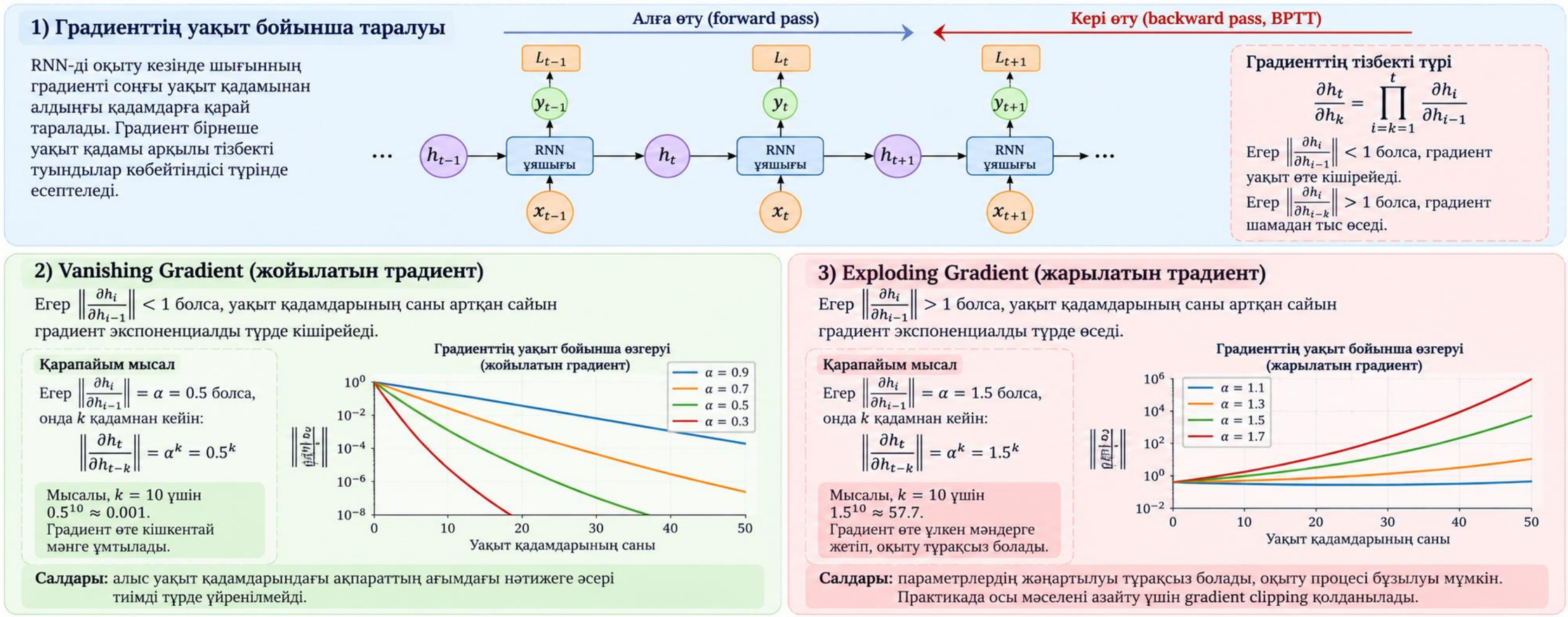
BPTT: қателерді уақыт бойынша кері тарату



BPTT градиенттерді есептейді; параметрлерді оптимизатор жаңартады.

Дереккөз: конспект, 4-бет, 7.2-сурет.

Ұзақ тәуелділіктер және градиент тұрақтылығы



Скалярлық үлгі — түсіндіру үшін; матрицалық градиенттің өсуі барлық бағытта бірдей емес.

Дереккөз: конспект, 5-бет, 7.3-сурет.

Градиент мәселелері және оларды азайту

Өшетін градиент

Скалярлық мысал:
$0{,}5^{10} \approx 0{,}001$.

Алыс қадамдардың әсерін үйрену қиындайды.

Шамадан тыс өсетін градиент

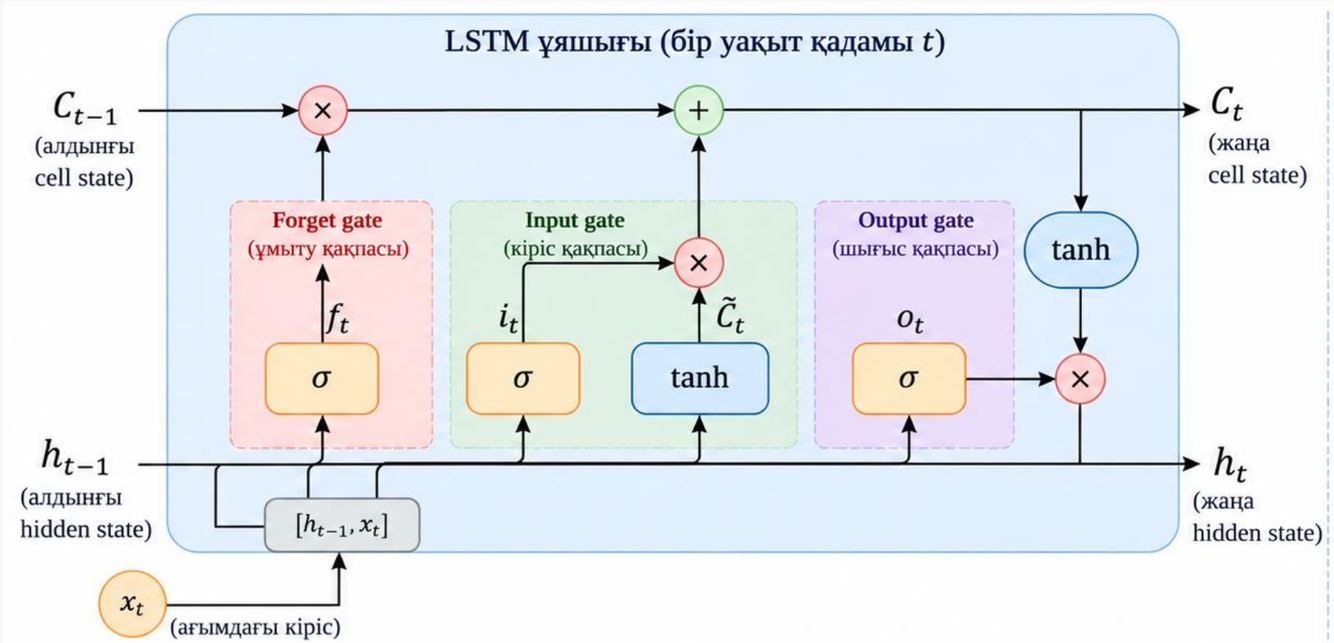
Скалярлық мысал:
$1{,}5^{10} \approx 57{,}7$.

Параметрлерді жаңарту тұрақсыз болуы мүмкін.

Gradient clipping: градиент нормасын c шегімен шектеу.
LSTM/GRU: ақпарат ағынын басқару арқылы ұзақ тәуелділіктерді үйренуді жеңілдету.

Clipping өшетін градиент мәселесін өздігінен шешпейді.

LSTM ұяшығы: күй және ақпарат ағыны



Белгілердің мағынасы

- x_t — уақыт t қадамындағы кіріс векторы
- h_{t-1} — алдынғы hidden state
- h_t — жаңа hidden state
- C_{t-1} — алдынғы cell state (ұзақ мерзімді жады)
- C_t — жаңа cell state
- f_t — ұмыту қақпасы
- i_t — кіріс қақпасы
- $\tilde{C}_t$ — жаңа кандидат ақпарат
- o_t — шығыс қақпасы
- σ — sigmoid функциясы ($0 \le \sigma(z) \le 1$)
- tanh — гиперболалық тангенс ($-1 \le \tanh(z) \le 1$)
- $\odot$ — элемент бойынша көбейту
- $[h_{t-1}, x_t]$ — векторлардың біріктірілуі (конкатенация)

Негізгі идея

- Cell state C_t – ұзақ мерзімді ақпаратты тасымалдайтын арна.
- Қақпалар (f_t, i_t, o_t) – ақпарат ағынын басқарады.
- LSTM қандай ақпаратты сақтау, қандай ақпаратты ұмыту және қандай ақпаратты шығару керектігін оқыту барысында үйренеді.

LSTM ұяшығының негізгі формулалары

1) Forget gate (ұмыту қақпасы)

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \quad (7.8)$$

Алдыңғы cell state-тің қандай бөлігін сақтау керектігін анықтайды.

2) Input gate (кіріс қақпасы)

$$i_t = \sigma\left(W_i[h_{t-1}, x_t] + b_i\right) \quad (7.9)$$

$$\tilde{C}_t = \tanh\left(W_C[h_{t-1}, x_t] + b_C\right) \quad (7.10)$$

Ағымдағы кірістен қандай жаңа ақпаратты cell state-ке жазу керектігін анықтайды.

3) Cell state (жадының жаңартылуы)

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (7.11)$$

Алдыңғы жадының сақталатын бөлігі және жаңа ақпараттың қосылуы.

4) Output gate (шығыс қақпасы)

$$o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right) \quad (7.12)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7.13)$$

Cell state-тегі ақпараттың қандай бөлігін hidden state ретінде шығару керектігін анықтайды.

Дереккөз: конспект, 6-бет, 7.4-сурет.

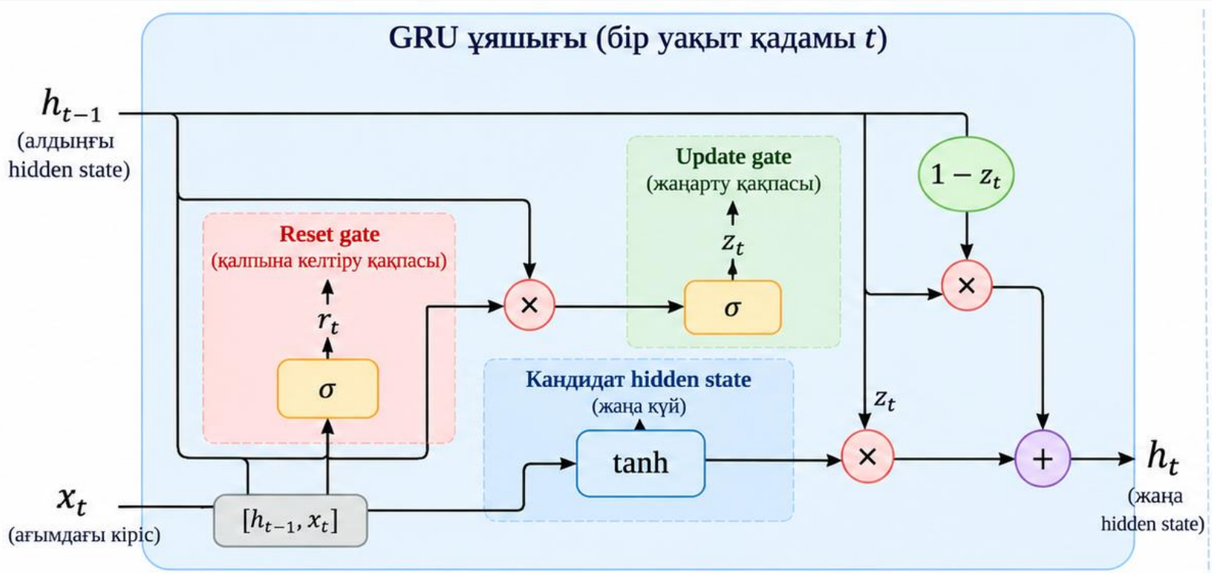
LSTM: жадыны сақтау, толықтыру және шығару

$s_t = [h_{t-1}, x_t]$; σ — sigmoid; $\odot$ — элементтік көбейту.

Ұмыту	$f_t = \sigma(W_f s_t + b_f)$
Жаңа ақпарат	$i_t = \sigma(W_i s_t + b_i)$; $\tilde{C}_t = \tanh(W_C s_t + b_C)$
Ұяшық күйі	$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$
Шығыс қақпасы	$o_t = \sigma(W_o s_t + b_o)$
Жасырын күй	$h_t = o_t \odot \tanh(C_t)$

C_t — ұяшық күйі; h_t — ағымдағы шығысқа және келесі қадамға берілетін жасырын күй.

GRU ұяшығы: ықшам қақпалы құрылым



Белгілердің мағынасы

x_t	— уақыт t қадамындағы кіріс векторы	W_r, W_z, W_h	— кіріс салмақтары (x_t)
h_{t-1}	— алдыңғы hidden state	U_r, U_z, U_h	— рекурренттік салмақтары (h_{t-1})
h_t	— жаңа hidden state	b_r, b_z, b_h	— ығысу векторлары
r_t	— reset gate (қалпына келтіру қақпасы)	σ	— sigmoid функциясы ($0 \le \sigma(z) \le 1$)
z_t	— update gate (жаңарту қақпасы)	tanh	— гиперболалық тангенс ($-1 \le \tanh(z) \le 1$)
$\tilde{h}_t$	— кандидат hidden state (жаңа күй)	$\odot$	— элемент бойынша көбейту
		$[h_{t-1}, x_t]$	— векторлардың біріктірілуі (конкатенация)

Негізгі идея

- GRU – LSTM-ге қарағанда қарапайым архитектура.
- Екі қақпа (r_t, z_t) hidden state арқылы ақпарат ағынын басқарады.
- Жеке cell state жоқ, алмыс ақпарат hidden state құрамында сақталады.
- Көптеген есептерде GRU LSTM-мен салыстырмалы нәтиже береді, бірақ параметрлер саны көбіне аз.

GRU ұяшығының негізгі формулалары

1) Reset gate (қалпына келтіру қақпасы)

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (7.15)$$

Жаңа кандидат күйді есептеу кезінде алдыңғы hidden state-тің қаншалықты пайдаланылатынын анықтайды.

2) Update gate (жаңарту қақпасы)

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (7.14)$$

Алдынғы hidden state-тегі ақпараттың қаншалықты сақталатынын және жаңа ақпараттың қаншалықты қосылатынын анықтайды.

3) Кандидат hidden state (жаңа күй)

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (7.16)$$

Ағымдағы кіріс пен (қажет жағдайда) алдынғы hidden state негізінде жаңа кандидат күй есептеледі.

4) Жаңа hidden state

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (7.17)$$

Алдынғы hidden state пен жаңа кандидат күй element бойынша біріктіріледі. z_t – жаңа ақпараттың үлесін басқарады.

Нақты есептеу үшін келесі слайдтағы теңдеулер қолданылады; сызбадағы жебелер шартты.

Дереккөз: конспект, 8-бет, 7.5-сурет.

GRU: reset және update қақпалары

Жеке cell state жоқ; ақпарат h_t күйі арқылы тасымалданады.

Update	$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$
Reset	$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$
Кандидат	$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$
Жаңа күй	$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$

$z_t \to 0$: бұрынғы күй сақталады.
$z_t \to 1$: жаңа кандидаттың үлесі артады.

Бұл конвенцияда z_t жаңа күйдің үлесін басқарады; әдебиетте кері белгілеу де кездеседі.

RNN, LSTM және GRU: салыстыру

Қасиет	RNN	LSTM	GRU
Күйлер	h_t	h_t және C_t	h_t
Қақпалар	Жоқ	Forget, input, output	Update, reset
Құрылым	Қарапайым	Күрделірек	Ықшам қақпалы
Ұзақ тәуелділік	Үйренуі қиындау	Жеңілдетілген	Жеңілдетілген

GRU бірдей күй өлшемінде көбіне LSTM-нен аз параметр қолданады.

Әмбебап үздік архитектура жоқ: таңдау валидациялық нәтижемен негізделеді.

Қолданбалы есеп: уақыттық қатарды болжау

Міндет: соңғы бес тәуліктің энергия тұтынуынан келесі мәнді болжау.

$$(x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t) \rightarrow h_t \rightarrow \hat{x}_{t+1}$$

$$\hat{x}_{t+1} = W_y h_t + b_y$$

Регрессиялық шығын: MSE = $(1/N)\,\Sigma_i (y_i - \hat{y}_i)^2$.
Терезе ұзындығы, шу, тренд және маусымдылық нәтижеге әсер етеді.

Тізбекті модельді оқытудың практикалық шарттары

Тізбек ұзындығы	Қажетті контекст пен есептеу шығынының теңгерімі.
Масштабтау	Масштабтау параметрлерін тек train деректерінде анықтау.
Бағалау	Болжауда уақыт ретін сақтау, болашақ ақпаратты кіріске қоспау.
Тұрақтылық	Gradient clipping, validation loss, dropout және early stopping.

Архитектураның өзі жеткіліксіз: деректерді дайындау мен бағалау тәртібі де шешуші.

Рекуррентттік модельдердің шектеулері

Рекуррентттік есептеу

h_t-ны табу үшін h_{t-1} қажет.

Ұзын тізбектерде уақыт қадамдарын толық қатар есептеу қиын.

Күйді сақтау және оқыту шығыны артады.

Attention және Transformer

Элементтер арасындағы байланысты рекуррентттік күйге сүйенбей модельдеуге мүмкіндік береді.

Бұл RNN-нің барлық қолданылуын жоққа шығармайды.

RNN, LSTM және GRU күй, жады және уақыттық тәуелділік ұғымдарын түсінуге негіз болады.

Бақылау сұрақтары

01 RNN-дегі жасырын күй не сақтайды және қалай жаңартылады?

02 Желіні уақыт бойынша ашу және ортақ салмақтар нені білдіреді?

03 BPTT пен оптимизатордың қызметтері қалай ажыратылады?

04 Градиенттің өшуі мен өсуі қалай пайда болады, clipping не үшін керек?

05 LSTM-дегі C_t, h_t және үш қақпаның қызметтері қандай?

06 GRU қақпалары қалай жұмыс істейді және ол LSTM-нен қалай ерекшеленеді?

Дереккөз және иллюстрациялар

Негізгі дереккөз

«7-дәріс. Рекурренттік нейрондық желілер: RNN, LSTM және GRU».
Пайдаланушы ұсынған дәріс конспектісі, 10 бет.

Конспектіден алынған барлық алты сурет бөлігі сақталған

7.1-сурет: RNN ұяшығы және уақыт бойынша ашылуы → 3–4-слайдтар.
7.2-сурет: BPTT → 6-слайд.
7.3-сурет: градиент мәселелері → 7-слайд.
7.4-сурет: LSTM → 9-слайд.
7.5-сурет: GRU → 11-слайд.

Есептеулерде конспектінің мәтіндік теңдеулері негізге алынды. Түпнұсқа схемалардың шартты тұстары оқытушы ескертпелерінде нақтыланған; белгілеулер біріздендірілген.