Нейрондық желілерді оқыту

Қателерді кері тарату алгоритмі

Шығын → градиент → параметрлерді жаңарту

Болжам

Forward

Шығын

Loss

Градиент

Backward

Жаңарту

Update

Негізгі сұрақ: желі салмақтар мен ығысулардың қажетті мәндерін қалай табады?

Дәрістің мақсаты және оқу нәтижелері

01 Бағалау

Болжам мен нақты мәннің айырмасын шығын арқылы сипаттау.

02 Түсіндіру

Туынды, градиент және оқыту жылдамдығының мағынасын ашу.

03 Есептеу

Тізбекті ереже арқылы параметрлердің градиенттерін табу.

04 Ажырату

Кері тарату мен параметрлерді жаңартудың қызметтерін ажырату.

Оқыту циклі: Forward → Loss → Backpropagation → Update

Алға таралудан оқытуға өту

$$x \rightarrow a^{(1)} \rightarrow a^{(2)} \rightarrow \ldots \rightarrow \hat{y}$$

Нақты мән: $y = 1$. Бастапқы болжам: $\hat{y} = 0{,}35$.

Қаншалықты қателестік?
Қай параметр қатеге қалай әсер етті?
Оны қай бағытта және қандай қадаммен өзгерту керек?

Оқыту — осы сұрақтарға туынды мен оңтайландыру арқылы жауап беру.

Оқыту деректері және модель параметрлері

$$D = \{(x^{(i)}, y^{(i)})\} , i = 1, \ldots, N$$

$$\hat{y}^{(i)} = F(x^{(i)}; W, b)$$

$x^{(i)}$ — белгілер; $y^{(i)}$ — белгілі дұрыс нәтиже.
W — салмақтар; b — ығысулар.
Оқыту осы параметрлерді деректерге сәйкес анықтайды.

Шығын функциясы: қатені санмен бағалау

$$L(y, \hat{y}) = \tfrac{1}{2}(y - \hat{y})^2$$

y = 1; ŷ = 0,8

L = ½(1 − 0,8)² = 0,02

y = 1; ŷ = 0,3

L = ½(1 − 0,3)² = 0,245

Осы шығын бойынша бірінші болжам нақты мәнге жақынырақ.

Оқытуды оңтайландыру есебі ретінде қарастыру

$$J(W, b) = (1/N) \Sigma_i L(y^{(i)}, \hat{y}^{(i)})$$

Мақсат: W, b бойынша J-ді азайту

Бір мысалдың шығыны мен барлық деректің орташа шығынын ажыратамыз.

Оқыту шығынының төмендеуі жаңа деректердегі сапаға өздігінен кепілдік бермейді.

Туынды және градиенттің мағынасы

∂J/∂w > 0

w-ны аздап арттыру J-ді өсіруге бейім.

Төмендеу бағыты: w-ны азайту.

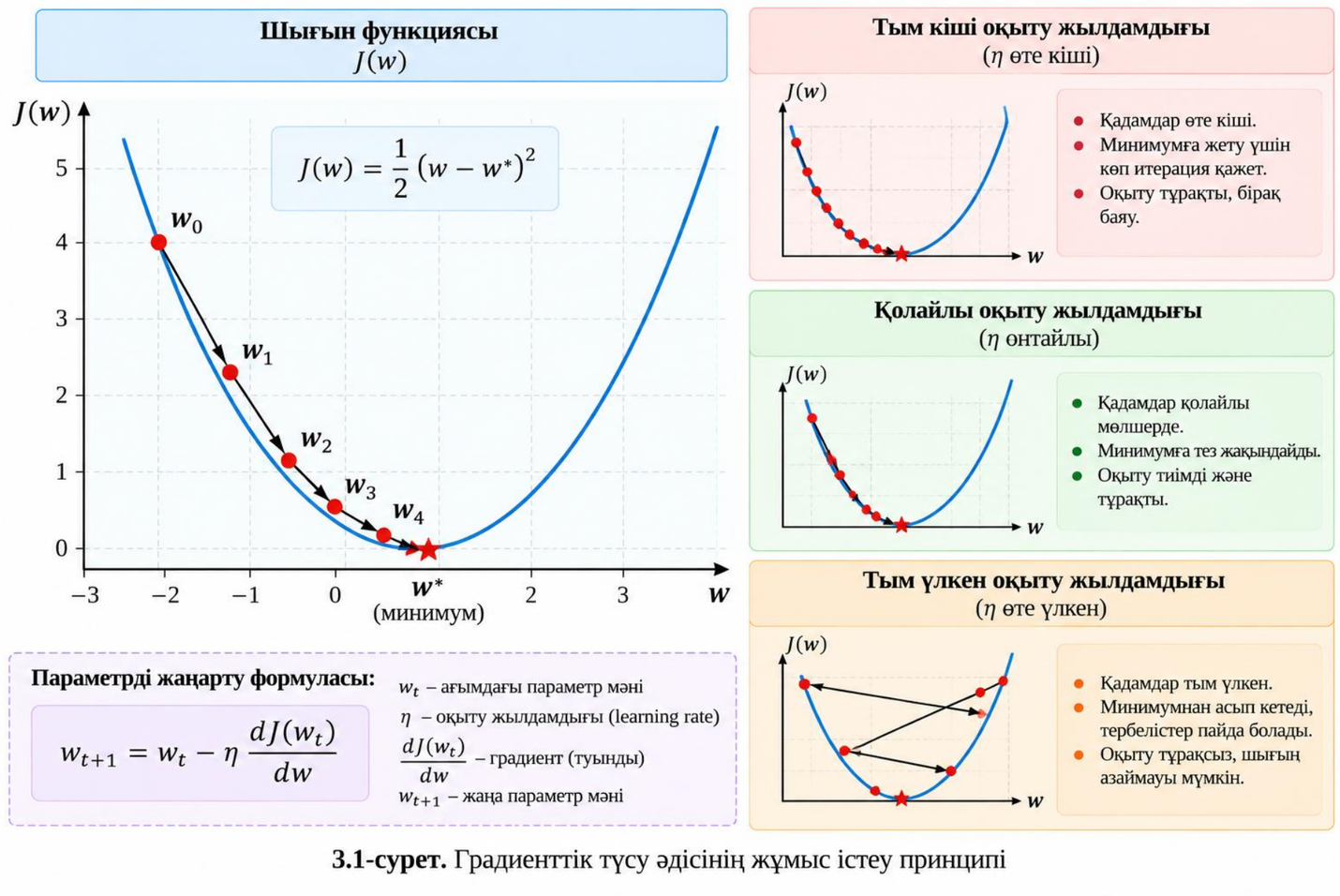
∂J/∂w < 0

w-ны аздап арттыру J-ді азайтуға бейім.

Төмендеу бағыты: w-ны арттыру.

$$\nabla J = (\partial J/\partial w_1, \ldots, \partial J/\partial w_n)^T$$

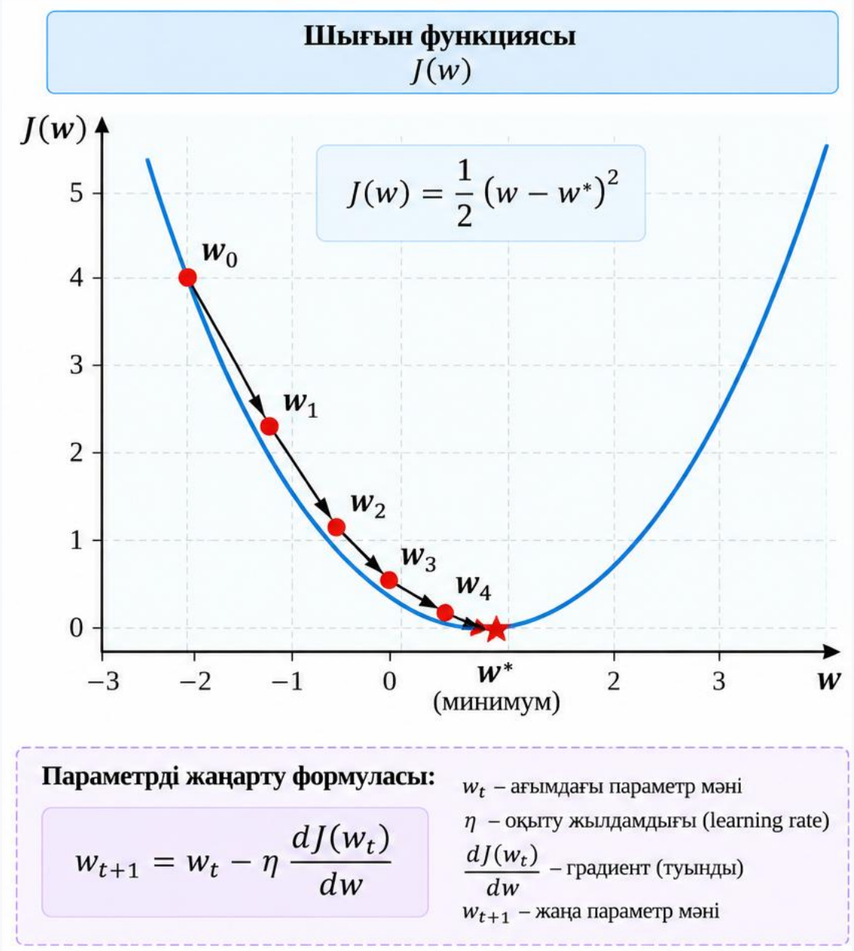
Градиенттік түсу: дәрістегі түпнұсқа сурет



3.1-сурет. Градиенттік түсу әдісінің жұмыс істеу принципі

Дереккөз: 3-дәріс конспектісі, 3-бет, 3.1-сурет. Түпнұсқа.

Градиентке қарсы бағытта қозғалу



1. Ағымдағы w мәнін аламыз.

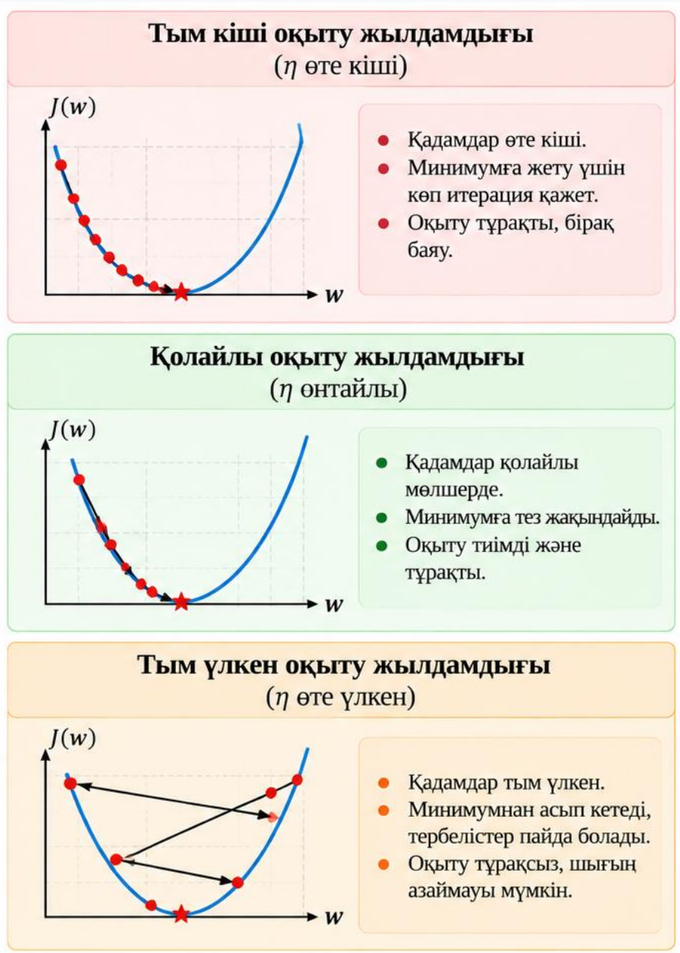
2. J-дің туындысын есептейміз.

3. Қарсы бағытта қадам жасаймыз.

4. Жаңа нүктеде қайталаймыз.

Дереккөз: конспект, 3-бет, 3.1-суреттің сол жақ бөлігі.

Оқыту жылдамдығы η нені өзгертеді?



Өте кіші η

Қадамдар ұсақ, оқыту баяу.

Қолайлы η

Жинақталу тиімдірек болуы мүмкін.

Тым үлкен η

Тербеліс не алшақтау қаупі бар.

Дереккөз: конспект, 3-бет, 3.1-суреттің оң жақ бөлігі.

Параметрлерді жаңарту ережесі

$$w_{жаңа} = w_{ескі} - \eta \, \partial J / \partial w$$

$$b_{жаңа} = b_{ескі} - \eta \, \partial J / \partial b$$

$\eta > 0$ — оқыту жылдамдығы.
Минус таңбасы градиентке қарсы бағытты береді.

w және b — параметрлер; η — оқытуды ұйымдастыратын гиперпараметр.

Тізбекті туынды ережесі

$$dy/dx = (dy/dz)(dz/dx)$$

Нейрон үшін: $z = wx + b$, $a = f(z)$, $J = L(y, a)$

$$∂J/∂w = (∂J/∂a) × f′(z) × x$$

Соңғы әсер = шығын сезімталдығы × активация туындысы × кіріс.

Кері тарату мен градиенттік түсу бірдей емес

Backpropagation

«Әр параметр шығынға қалай әсер етеді?»

∂J/∂W және ∂J/∂b есептейді.

Параметрді өзі жаңартпайды.

Gradient Descent

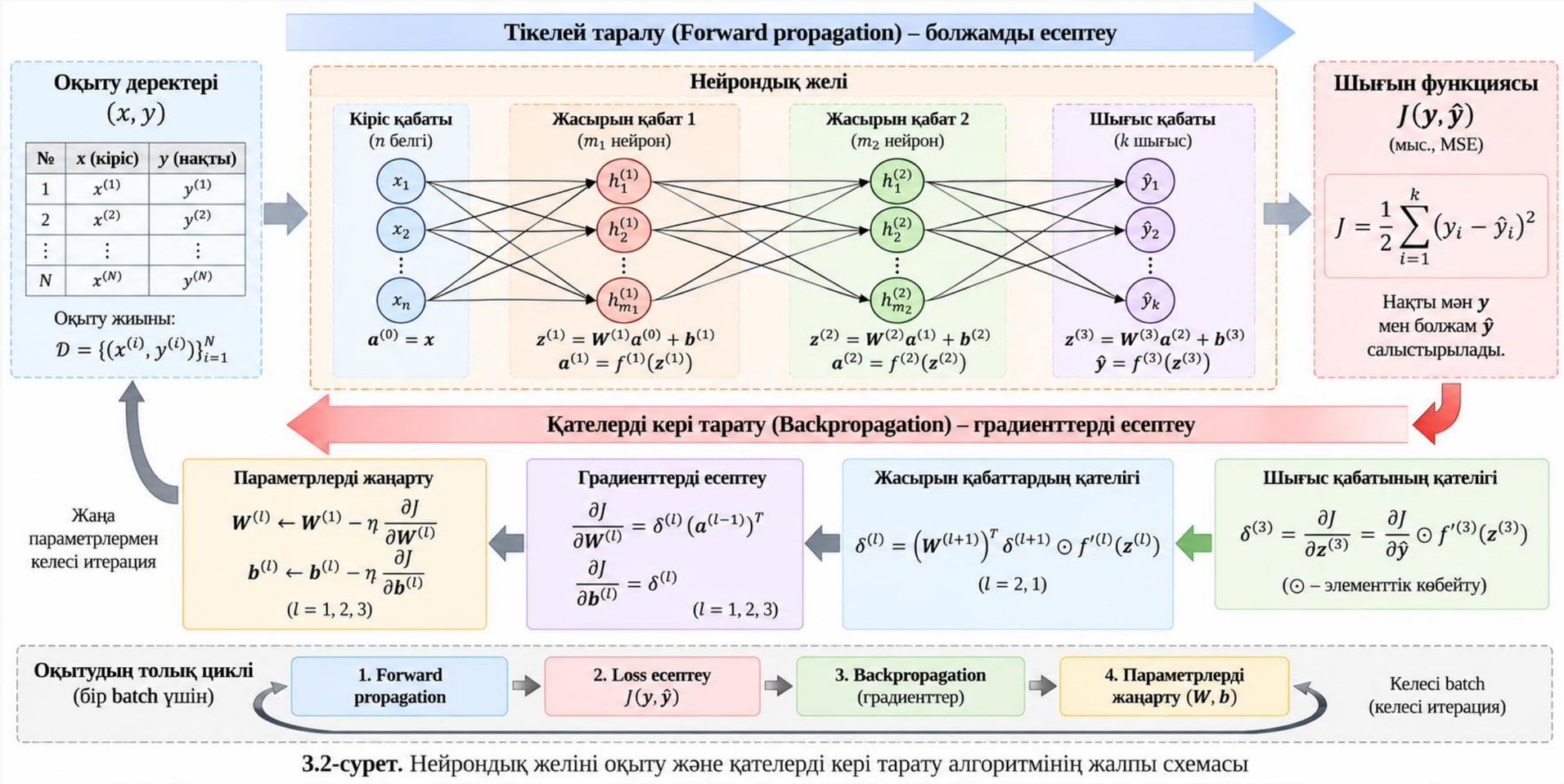
«Параметрді қалай өзгертеміз?»

Есептелген градиентті пайдаланады.

$W \leftarrow W - \eta \nabla J$

Алдымен градиентті есептейміз, содан кейін параметрлерді жаңартамыз.

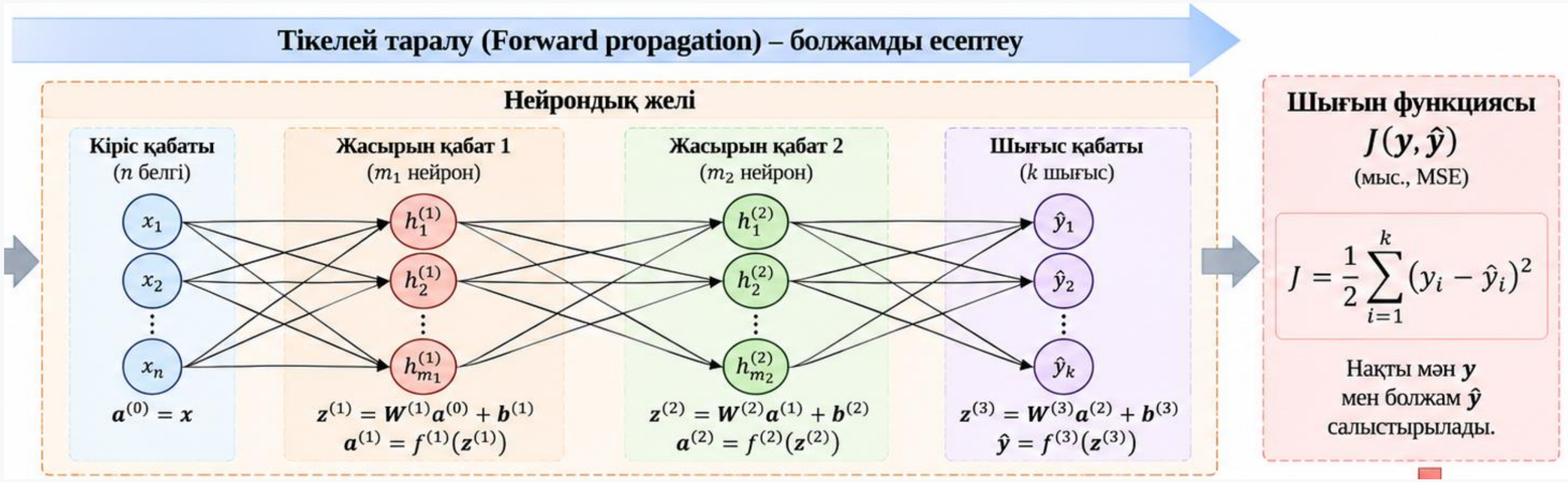
Оқыту және кері тарату: толық схема



3.2-сурет. Нейрондық желіні оқыту және қателерді кері тарату алгоритмінің жалпы схемасы

Дереккөз: 3-дәріс конспектісі, 5-бет, 3.2-сурет. Түпнұсқа.

Алға таралу: нені сақтаймыз?



Әр қабатта: $z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$, $a^{(l)} = f^{(l)}(z^{(l)})$.

Кері тарату үшін аралық z және a мәндері сақталады.

Дереккөз: конспект, 5-бет, 3.2-суреттің жоғарғы бөлігі.

Шығыс нейронының градиенті

$$J = \tfrac{1}{2}(y - \hat{y})^2,\ \hat{y} = f(z),\ z = wa + b$$

$$\partial J/\partial w = (\hat{y} - y)\, f'(z)\, a$$

$$\partial J/\partial b = (\hat{y} - y)\, f'(z)$$

Ауытқу × активация сезімталдығы × байланысқа келетін кіріс.

Жасырын қабатқа градиент қалай беріледі?

$$\delta^{(l)} = \partial J / \partial z^{(l)}$$

$$\delta^{(l)} = \left[(W^{(l+1)})^{\top} \delta^{(l+1)} \right] \odot f'^{(l)}(z^{(l)})$$

Кейінгі қабат сигналдары салмақтар арқылы кері біріктіріледі.
Содан кейін ағымдағы активацияның туындысы ескеріледі.

⊙ — сәйкес элементтерді көбейту; δ — шығынның z-ке сезімталдығы.

Салмақтар мен ығысулардың градиенттері

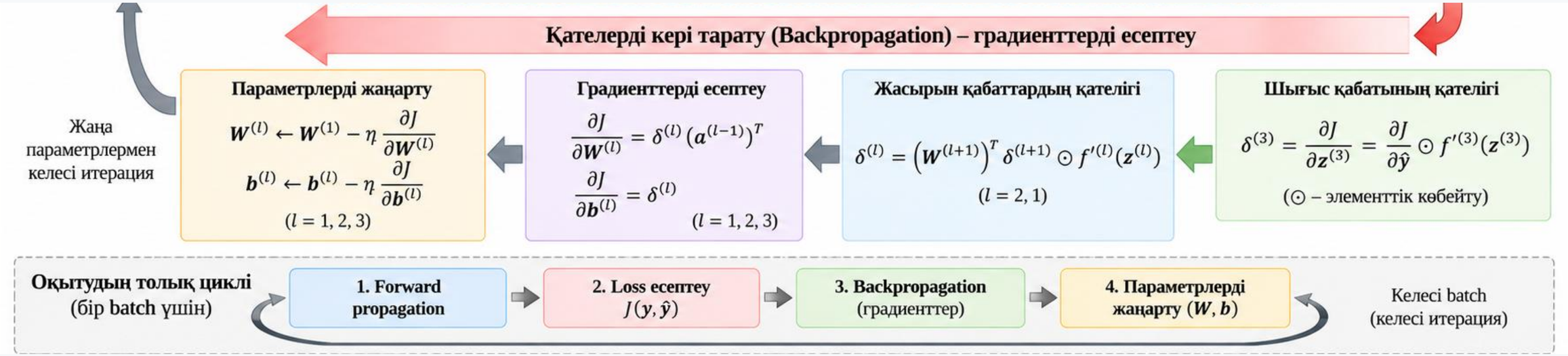
$$\partial J/\partial W^{(l)} = \delta^{(l)} \, (a^{(l-1)})^{T}$$

$$\partial J/\partial b^{(l)} = \delta^{(l)}$$

Өлшемдер: $(m \times 1)(1 \times n) = m \times n$.
Нәтиже W матрицасының өлшеміне сәйкес келеді.

Mini-batch үшін градиенттер шығынның анықтамасына сай орташаланады.

Толық цикл: градиенттен жаңа параметрге



Дереккөз: конспект, 5-бет, 3.2-суреттің төменгі бөлігі.

Сандық мысал: берілген шамалар

Деректер

Кіріс: $x = 2$

Дұрыс жауап: $y = 3$

Модель

$w = 0{,}5$
$b = 0$

$f(z) = z$
$\hat{y} = wx + b$

Оқыту

$J = \frac{1}{2}(y - \hat{y})^2$

$\eta = 0{,}1$

Бір жаңарту жасаймыз.

Табу керек: градиенттер, жаңа w мен b, жаңа болжам және шығын.

1–2-қадам: болжам және бастапқы шығын

$$z = 0{,}5 \times 2 + 0 = 1,\ \hat{y} = 1$$

$$\text{Jeckі} = ½(3 - 1)^2 = 2$$

Қазір параметрлер әлі өзгерген жоқ.

Forward + Loss тек бастапқы қатені бағалады; келесі қадам — туындылар.

3-қадам: градиенттерді есептеу

$$\partial J/\partial \hat{y} = \hat{y} - y = 1 - 3 = -2$$

$$\partial \hat{y}/\partial z = 1,\ \partial z/\partial w = x = 2,\ \partial z/\partial b = 1$$

$$\partial J/\partial w = (-2) \times 1 \times 2 = -4$$

$$\partial J/\partial b = (-2) \times 1 \times 1 = -2$$

4-қадам: параметрлерді жаңарту

$$w_{жаңа} = 0{,}5 - 0{,}1 \times (-4) = 0{,}9$$

$$b_{жаңа} = 0 - 0{,}1 \times (-2) = 0{,}2$$

Теріс градиенттерге байланысты екі параметр де артты.

«Минус теріс сан» оң өзгеріс береді: жаңартудың таңбасын тексеріңіз.

5-қадам: жаңа болжамды тексеру

$$\hat{y}_{\text{жаңа}} = 0{,}9 \times 2 + 0{,}2 = 2$$

$$J_{\text{жаңа}} = \tfrac{1}{2}(3 - 2)^2 = 0{,}5$$

Болжам

1 → 2 (мақсат: 3)

Шығын

2 → 0,5 (4 есе азайды)

Batch, iteration және epoch

Batch

Бір қадамда өңделетін мысалдар тобы.

Batch size = 100.

Iteration

Параметрлерді бір рет жаңарту қадамы.

Осы мысалда әр batch-тен кейін.

Epoch

Бүкіл оқыту жиынының бір толық өтуі.

N = 10 000 мысал.

10 000 / 100 = 100 batch; әр batch-ке бір жаңарту болса, бір epoch = 100 iteration.

Градиенттік түсудің үш нұсқасы

Batch GD

Градиент бүкіл оқыту жиыны бойынша есептеледі.

Бір жаңарту қымбат болуы мүмкін.

Stochastic GD

Әр жеке мысалдан кейін жаңарту жасалады.

Градиент бағасы құбылмалырақ.

Mini-batch GD

Шағын топтың орташа градиенті қолданылады.

Есептеу тиімділігі мен тұрақтылықтың теңгерімі.

Нұсқалардың айырмасы — жаңартуға дейін градиент есептелетін мысалдар саны.

Оқытудың толық циклін сөзбен түсіндіру

1 Инициализация	W және b бастапқы мәндері
2 Forward + Loss	Болжам және шығын
3 Backpropagation	Барлық параметрдің градиенті
4 Update	Жаңа W, b → келесі batch

Бір epoch аяқталғанда деректер қайта өңделуі мүмкін; цикл бірнеше рет қайталанады.

Бақылау сұрақтары

01 Оқыту кезінде қандай параметрлер өзгереді?

02 L және J функцияларының айырмасы қандай?

03 Градиент нені көрсетеді және неге оған қарсы жүреміз?

04 Тізбекті ереже кері таратуда қалай қолданылады?

05 Backpropagation мен Gradient Descent қалай ажыратылады?

06 Epoch, batch және iteration ұғымдары қалай байланысады?

Негізгі нәтижелер

$$W \leftarrow W - \eta \, \partial J/\partial W,\ b \leftarrow b - \eta \, \partial J/\partial b$$

Шығын болжам қатесін сандық сипаттайды.
Градиент параметрлерге сезімталдықты көрсетеді.
Кері тарату градиенттерді тиімді есептейді.
Оңтайландыру әдісі параметрлерді жаңартады.
Оқыту жылдамдығы қадам мөлшерін анықтайды.

Есте сақтаңыз: алға таралу → шығын → кері тарату → жаңарту.

Дереккөз және суреттердің шығу тегі

Негізгі материал

«3-дәріс. Нейрондық желілерді оқыту. Қателерді кері тарату алгоритмі». Пайдаланушы ұсынған дәріс конспектісі, 9 бет.

Презентацияда қолданылған түпнұсқа суреттер

3.1-сурет — градиенттік түсу және оқыту жылдамдығы, 3-бет.
3.2-сурет — оқыту және кері тарату алгоритмінің схемасы, 5-бет.

Толық суреттер 8 және 14-слайдтарда берілген.
Үлкейтілген бөліктер 9, 10, 15 және 19-слайдтарда көрсетілген.
Формулалар мен сандық мысал дәріс мазмұны бойынша сақталған.