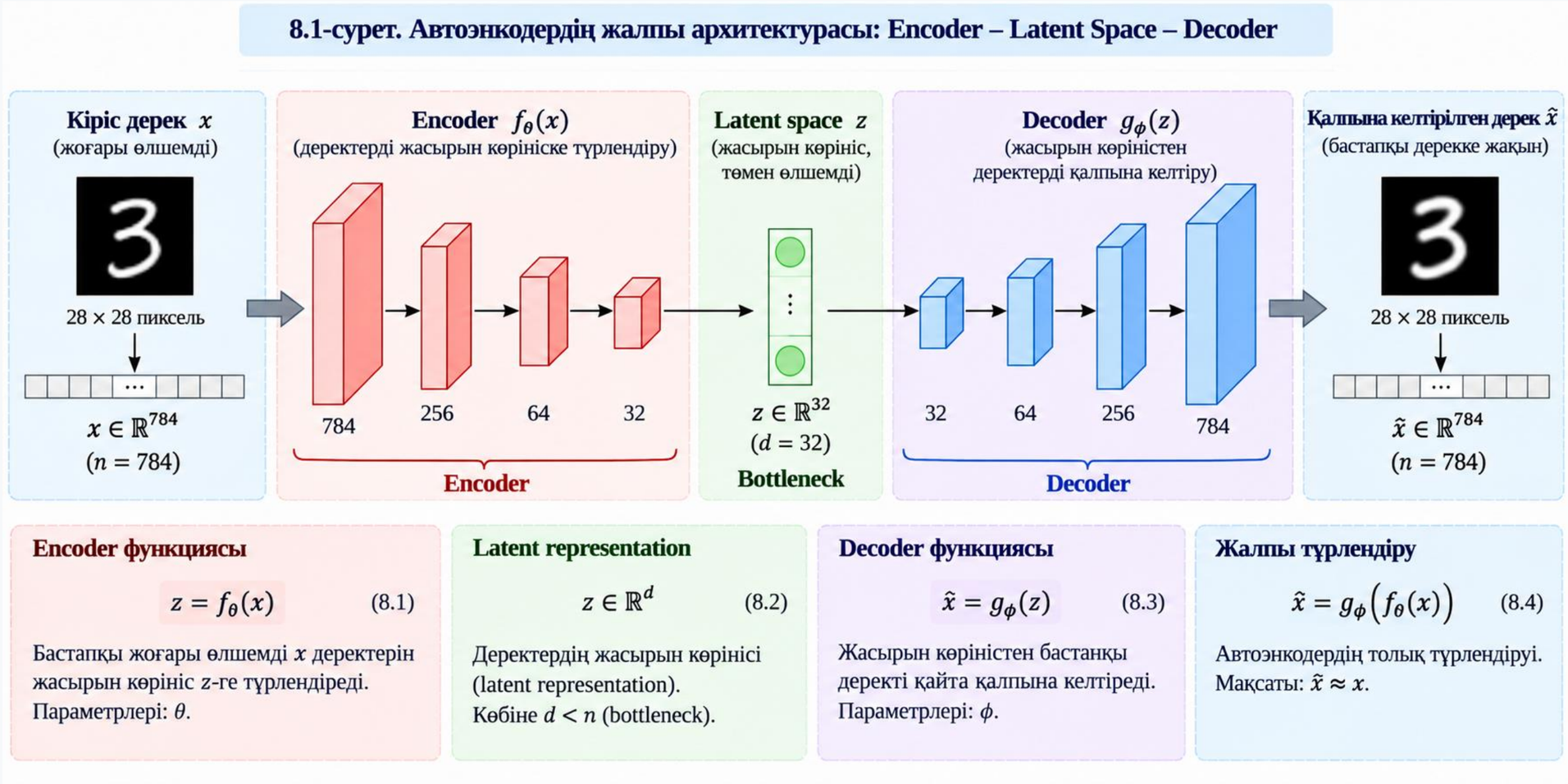
Автоэнкодерлер және деректердің жасырын белгілерін ұсыну

Деректерді қайта құру арқылы пайдалы жасырын көріністі қалай үйренуге болады?

Дәрістің негізгі міндеттері

- Encoder, latent space және Decoder қызметтерін түсіндіру.
- Reconstruction loss пен архитектуралық шектеулерді талдау.
- Sparse, denoising және аномалияларды анықтау тәсілдерін салыстыру.

Автоэнкодердің жалпы архитектурасы



Дереккөз: 8-дәріс конспектісі, 2-бет, 8.1-сурет.

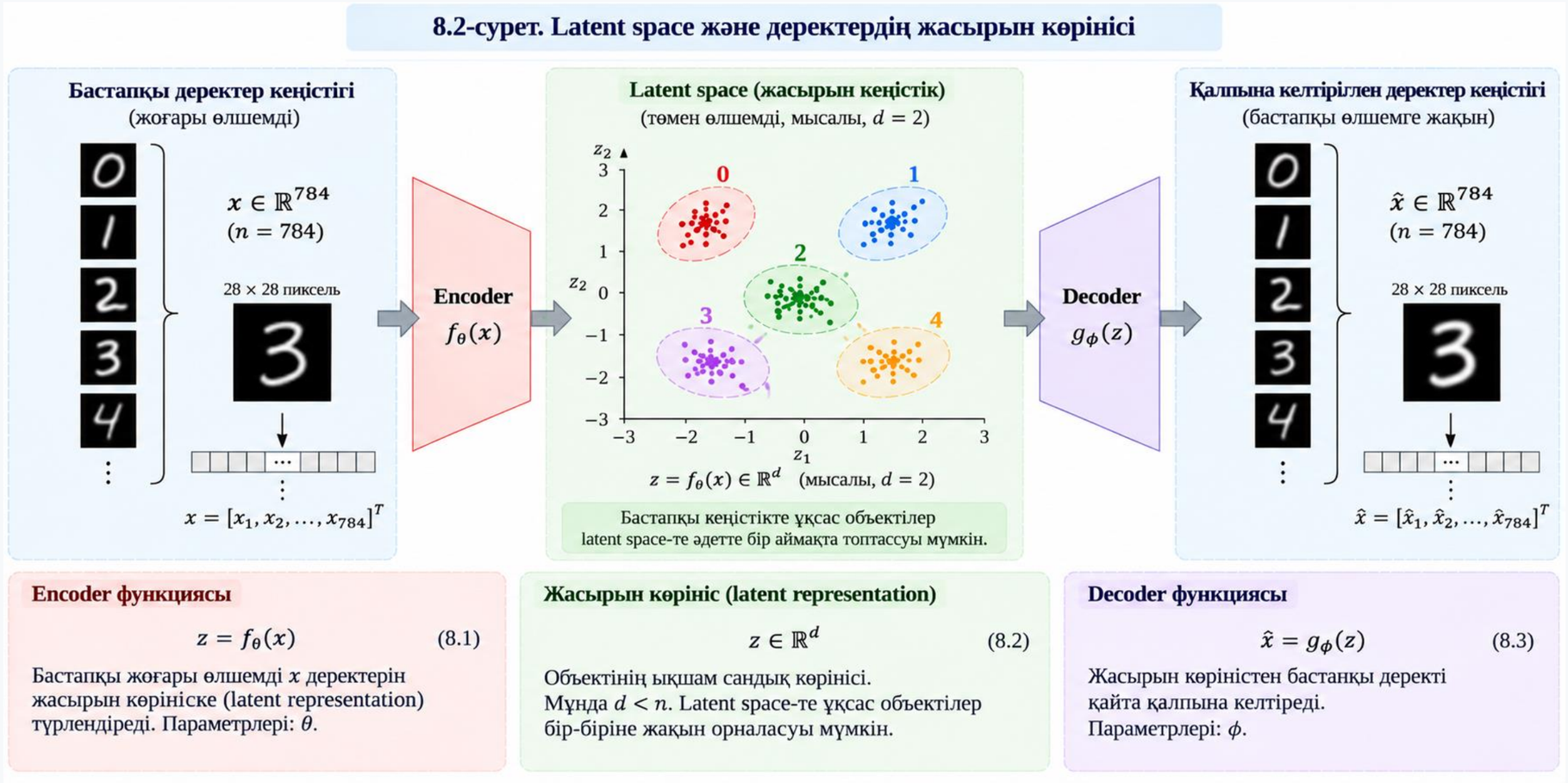
Encoder және Decoder: математикалық модель

$$z = f_\theta(x),\ \hat{x} = g_\varphi(z)$$

$$\hat{x} = g_\varphi(f_\theta(x))$$

Encoder: $x \in \mathbb{R}^n \to z \in \mathbb{R}^d$.
Decoder: $z \in \mathbb{R}^d \to \hat{x} \in \mathbb{R}^n$.
Классикалық оқыту жұбы: $x \to x$; класс белгілері міндетті емес.

Latent space: жасырын көрініс кеңістігі



Кластерлену — тұжырымдамалық үлгі; кәдімгі автоэнкодер оған кепілдік бермейді.

Дереккөз: конспект, 3-бет, 8.2-сурет.

Reconstruction loss және оқыту мақсаты

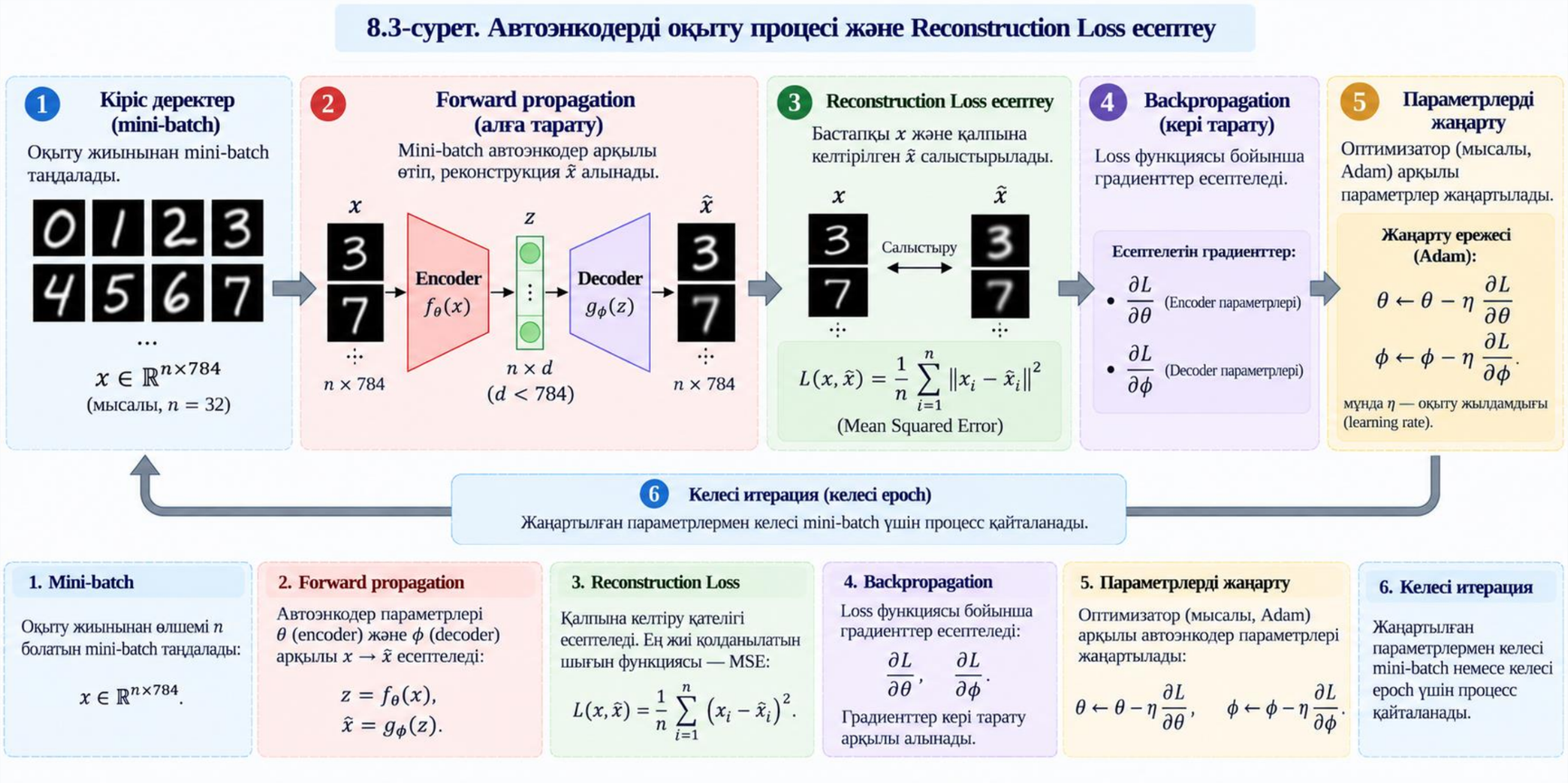
$$L(x, \hat{x}) = (1/n)\,\Sigma_j (x_j - \hat{x}_j)^2$$

$$J(\theta, \varphi) = (1/N)\,\Sigma_i\, L(x^{(i)}, \hat{x}^{(i)})$$

n — объектінің белгілер саны; N — оқыту объектілерінің саны.
Мақсат: θ және φ бойынша J(θ, φ) функциясын азайту.

Қалпына келтіру сапасы мен жасырын көріністің кейінгі есепке пайдалылығы бір ұғым емес.

Автоэнкодерді оқыту циклі



Нақтылау: суреттегі формула — градиенттік түсу, Adam-ның толық ережесі емес. Келесі mini-batch — жаңа итерация; ол міндетті түрде жаңа epoch емес.

Дереккөз: конспект, 4-бет, 8.3-сурет. Түпнұсқа сақталған.

Bottleneck: өлшемділікті азайту және ақпаратты сақтау

Encoder: 784 → 256 → 64 → 32.
Decoder: 32 → 64 → 256 → 784.

$$r = d/n = 32/784 \approx 0{,}041 = 4{,}1\%$$

Код тым шағын болса

Маңызды ақпарат жоғалып, қайта құру қатесі өсуі мүмкін.

Код тым кең болса

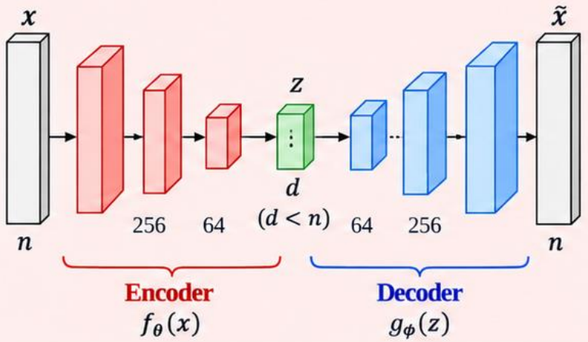
Қосымша шектеусіз кірісті көшіруге жақын шешім туындауы мүмкін.

4,1% — өлшемдер қатынасы; файл көлемінің нақты сығылу коэффициенті емес.

Undercomplete, Overcomplete және Sparse

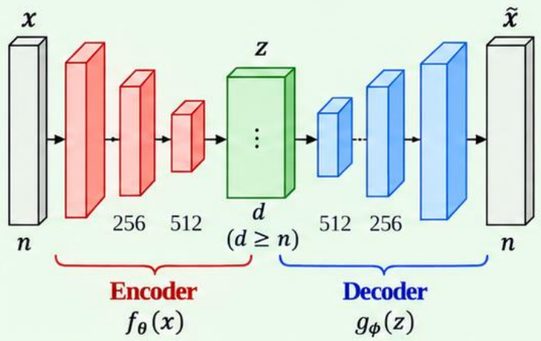
8.4-сурет. Автоэнкодердің негізгі түрлері: Undercomplete, Overcomplete және Sparse

1. Undercomplete Autoencoder
(тар bottleneck)



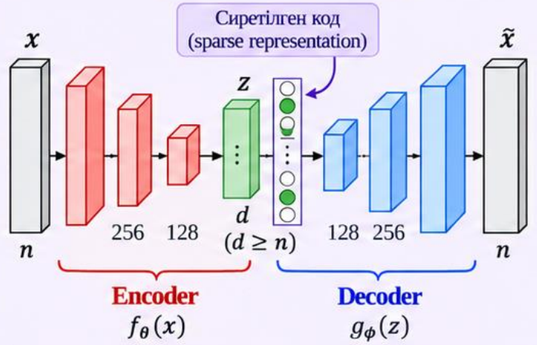
- Жасырын кеңістіктің өлшемі кіші: $d < n$.
- Деректерді қысылған түрде (компрессия) ұсынады.
- Маңызды құрылымды сақтауға үйренеді.

2. Overcomplete Autoencoder
(кең bottleneck)



- Жасырын кеңістіктің өлшемі бастапқыдан үлкен немесе тең: $d \geq n$.
- Шешім ішкі бірлікті бейнелеуге (identity mapping) ауып кетуі мүмкін: $\tilde{x} \approx x$.
- Қосымша шектеулер (мысалы, sparsity, denoising) қажет.

3. Sparse Autoencoder
(сиретілген автоэнкодер)



- Жасырын кеңістікте көптеген нейрондар болады ($d \geq n$), бірақ бір мезетте аз бөлігі ғана белсенді.
- Шығару үшін sparsity-регуляризация қолданылады:

$$L_{total} = L(x, \tilde{x}) + \lambda \Omega(z),$$

$$\Omega(z) = \sum_{j=1}^{d} |z_j| \quad \text{немесе} \quad KL(z_j \parallel \rho).$$

Сипаттама	Undercomplete	Overcomplete	Sparse
Latent space өлшемі	$d < n$	$d \geq n$	$d \geq n$ (бірақ сиретілген)
Негізгі идея	Компрессия, маңызды құрылымды сақтау	Жоғары өлшемді код, identity mapping қаупі	Көп нейрон, сиретілген белсенділік
Қолдану	Өлшемді азайту, визуализация	Қосымша шектеулермен (денойзинг, sparsity)	Маңызды факторларды бөліп алу, интерпретация, anomaly detection.

Нақтылау: Sparse Autoencoder үшін d ≥ n болуы міндетті емес; негізгі шектеу — белсенділік сиректігі.

Дереккөз: конспект, 6-бет, 8.4-сурет.

Өлшем шектеуі және сиректік шектеуі

Тәсіл	Негізгі шарт немесе мақсат
Undercomplete	d < n: төмен өлшемді код қалыптастыру.
Кең не тең код	d ≥ n: пайдалы көрініс үшін қосымша шектеу қажет болуы мүмкін.
Sparse	Жасырын белсенділіктердің аз бөлігін пайдалану.

$$J_{sparse} = J_{reconstruction} + \lambda R_{sparsity}$$

λ — сиректік пен қалпына келтіру сапасының арақатынасын реттейді.

Denoising Autoencoder: шулы кіріс, таза мақсат

$$\tilde{x} = x + \varepsilon \rightarrow \text{Encoder} \rightarrow z \rightarrow \text{Decoder} \rightarrow \hat{x}$$

Модель кірісі

Бұрмаланған немесе шулы $\tilde{x}$ объектісі.

Қалпына келтіру мақсаты

Бастапқы таза x объектісі.
Шығын: $L(x, \hat{x})$, $L(\tilde{x}, \hat{x})$ емес.

Деректегі тұрақты құрылымды сақтау және бұрмалануға төзімді көрініс үйрену көзделеді.

PCA және автоэнкодер: ұқсастық пен айырмашылық

PCA

Дисперсияны барынша сақтайтын сызықтық ішкі кеңістікке проекциялайды.

Классикалық сызықтық базалық әдіс.

Автоэнкодер

Бейсызық активациялар арқылы бейсызық түрлендірулерді үйрене алады.

Архитектура мен оқыту нәтижесіне тәуелді.

Орталықтандырылған деректер, сызықтық модель және MSE жағдайында оңтайлы төмен өлшемді ішкі кеңістік PCA кеңістігімен сәйкес келуі мүмкін.

Код координаталары PCA компоненттерінің мәндерімен міндетті түрде бірдей емес.

Қайта құру қатесі арқылы аномалияны анықтау

1. Автоэнкодерді қалыпты объектілерде оқыту.
2. Жаңа объектінің reconstruction error мәнін есептеу.
3. Қатені негізделген τ шегімен салыстыру.

$$E(x) = (1/n)\,\Sigma_j(x_j - \hat{x}_j)^2;\ E(x) > \tau$$

Мысалдар: өндірістік датчиктер, жабдық күйі, ерекше кескіндер.

Жоғары қате аномалияны автоматты дәлелдемейді; шек validation мен есеп контекстіне сүйенеді.

Encoder-ді белгі шығару үшін пайдалану

x → оқытылған Encoder → z → кейінгі модель

Қолданылуы

Жіктеу немесе регрессия.
Өлшемділікті азайту.
2D/3D көріністерді зерттеу.

Қалай бағаланады?

Қалпына келтіру қатесі.
Кейінгі есептегі модель сапасы.
Жаңа деректерге жалпылау қабілеті.

Encoder сапасын қолданудың соңғы мақсатына сәйкес тексеру қажет.

Шектеулер және генеративті модельдерге өту

Классикалық автоэнкодер

Мақсат: x → z → x̂.

Жасырын кеңістік құрылымы мен кодтың түсіндірмелілігіне кепілдік жоқ.

Нәтиже оқыту үлестіріміне тәуелді.

Генеративті міндет

Мақсат: z_жаңа → Decoder → x_жаңа.

Кездейсоқ z-тен мағыналы объект алу үшін жасырын кеңістікті арнайы ұйымдастыру қажет.

Кәдімгі автоэнкодердің жақсы реконструкциясы еркін генерациялау қабілетін білдірмейді.

Бақылау сұрақтары

01 Encoder, latent space және Decoder қандай қызмет атқарады?

02 Оқыту мақсаты неге кірістің өзінен құрылады?

03 Bottleneck және sparsity қандай әртүрлі шектеулер енгізеді?

04 Denoising Autoencoder-де кіріс пен мақсат қалай ажыратылады?

05 PCA-мен байланыс және reconstruction error арқылы аномалияны табу қалай түсіндіріледі?

06 Неліктен төмен reconstruction loss пайдалы кодқа немесе сәтті генерацияға кепіл емес?

Дереккөз және иллюстрациялар

Негізгі дереккөз

«8-дәріс. Автоэнкодерлер және деректердің жасырын белгілерін ұсыну». Пайдаланушы ұсынған конспект, 9 бет.

Төрт түпнұсқа схема сақталды

8.1-сурет: жалпы архитектура, 2-бет → 2-слайд.
8.2-сурет: latent space, 3-бет → 4-слайд.
8.3-сурет: оқыту циклі, 4-бет → 6-слайд.
8.4-сурет: автоэнкодер түрлері, 6-бет → 8-слайд.

Қысқа нақтылаулар: GD және Adam айырмасы; итерация мен epoch; Sparse Autoencoder өлшеміне қойылатын шарт; өлшемді азайту мен файл көлемін сығудың айырмашылығы.