

Rețele neuronale și deep learning

Cursul 8: Modele generative: autoencodere, VAE, GAN, difuzie



Prof. dr. Daniel Traian PELE

Academia de Studii Economice din București (ASE), Departamentul de Statistică și Econometrie
Academia Română, Institutul de Prognoză Economică (IPE)
IDA Institute Digital Assets ▪ AI4EFin ▪ MSCA Digital Finance

Obiectivele cursului

- Diferența dintre un model discriminativ $p(y \mid \mathbf{x})$ și un model generativ $p(\mathbf{x})$
  - De ce finanțele au nevoie de amîndouă
- Derivarea și antrenarea a patru familii de modele
  - Autoencoderul (AE), autoencoderul variațional (VAE), rețeaua generativă adversarială (GAN) și modelul probabilistic de difuzie (DDPM)
- Aplicarea lor pe date reale
  - Detectarea fraudei cu carduri, imagini de îmbrăcăminte în formatul cifrelor Modified National Institute of Standards and Technology (MNIST), randamentele S&P 500 și curba randamentelor din zona euro, fiecare comparat cu un model clasic
- Evaluarea datelor sintetice
  - Cozile distribuției, gruparea volatilității, măsuri de risc precum valoarea la risc (VaR) și ce nu poate reproduce un generator

Abrevieri

AE	autoencoder (rețea de autocodificare)
ASE	Academia de Studii Economice din București
DDPM	model probabilistic de difuzie cu eliminarea zgomotului (denoising diffusion)
ELBO	marginea inferioară a verosimilității (evidence lower bound)
ES	pierderea așteptată în coadă (expected shortfall)
GAN	rețea generativă adversarială
IPE	Institutul de Prognoză Economică, Academia Română
JS	divergența Jensen–Shannon
KL	divergența Kullback–Leibler
LLM	model mare de limbaj
MLP	perceptron multistrat
MNIST	cifrele scrise de mînă ale Modified National Institute of Standards and Technology
PCA	analiza componentelor principale
PR-AUC	aria de sub curba precizie-recall
ReLU	unitate liniară rectificată (rectified linear unit)
ROC	caracteristica de operare a receptorului (receiver operating characteristic)
ROC-AUC	aria de sub curba ROC
TSTR	antrenare pe date sintetice, test pe date reale (train on synthetic, test on real)
VAE	autoencoder variațional
VaR	valoarea la risc (Value-at-Risk)
WGAN	GAN Wasserstein

Cuprins

- Principiile modelării generative
- Autoencodere: detectarea anomaliilor la fraude cu carduri
- Autoencodere variaționale
- Rețele generative adversariale
- Modele de difuzie
- Modelele generative în practică
- Concluzii

Un model generativ învață cum apar datele

- **Pe scurt:** Un clasificator învață granița dintre clase; un model generativ învață toată distribuția datelor.

$$\underbrace{p(y \mid \mathbf{x})}_{\text{discriminativ}} \quad \text{vs.} \quad \underbrace{p(\mathbf{x}) \text{ sau } p(\mathbf{x}, y) = p(\mathbf{x} \mid y)\, p(y)}_{\text{generativ}}$$

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log p_\theta(\mathbf{x}_i)$$

- **unde**
  - $\mathbf{x}$ = o observație (tranzacție, imagine, zi de randamente, curbă)
  - y = o etichetă
  - p_θ = densitatea modelului, estimată prin verosimilitate maximă.

- **Patru utilizări în finanțe**
  - Detectarea anomaliilor: $p(\mathbf{x})$ mic = suspect (fraudă)
  - Scenarii pentru teste de stres și măsuri de risc
  - Date sintetice partajate fără înregistrări despre clienți

- **În practică:** un clasificator ordonează clienții, un model generativ arată și ce s-ar putea întîmpla.

(Bishop & Bishop, 2024, ch. 16)

Modele cu variabile latente

- **Pe scurt:** Unsprezece randamente sau 784 de pixeli sînt determinate de cîțiva factori ascunși; un model cu variabile latente formalizează acest lucru.

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})\, d\mathbf{z}, \qquad \mathbf{z} \in \mathbb{R}^k,\ k \ll d$$

liniar-gaussian: $\mathbf{x} = W\mathbf{z} + \boldsymbol{\mu} + \varepsilon,\ \mathbf{z} \sim \mathcal{N}(0, I),\ \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$

- **unde**
  - $\mathbf{z}$ = factorii latenți, în număr de k
  - d = dimensiunea datelor
  - Cazul liniar-gaussian este analiza probabilistă a componentelor principale (PCA), cu verosimilitate în formă închisă
  - Cu un decodor neuronal integrala nu mai are formă închisă.

- **Curba randamentelor**
  - Nivel, pantă, curbură: trei factori explică majoritatea mișcărilor zilnice
- **Prețul neliniarității**
  - VAE: o margine inferioară pentru $\log p$; GAN: nicio verosimilitate; difuzia: o margine construită pas cu pas

- **În practică:** la evaluarea unui model, întrebați care parte este povestea latentă și care este zgomot estimat.

(Bishop & Bishop, 2024, sec. 16.2)

Patru familii de modele generative

Familia	Ce se optimizează	Verosimilitate	Generarea	În acest curs
AE	eroarea de reconstrucție	nu	nu este generator	frauda cu carduri ca anomalie
VAE	marginea inferioară a verosimilității	margine inferioară	o trecere, imagini neclare	imagini de îmbrăcăminte; curbe ale randamentelor
GAN	joc contra unui discriminator	nu	o trecere, imagini clare	randamentele zilnice S&P 500
DDPM	eroarea de eliminare a zgomotului	margine inferioară	mulți pași, fidel	curbele randamentelor din zona euro

- **Nimic nu este gratuit**: verosimilitatea, calitatea eșantioanelor și viteza rareori vin împreună
  - Alegerea depinde de utilizare
- **Fluxurile de normalizare** (verosimilitate exactă, rețele inversabile) completează tabloul
  - Vezi Bishop & Bishop (2024), cap. 18

(Bishop & Bishop, 2024, ch. 17–20)

Cuprins

- Principiile modelării generative
- Autoencodere: detectarea anomaliilor la fraude cu carduri
- Autoencodere variaționale
- Rețele generative adversariale
- Modele de difuzie
- Modelele generative în practică
- Concluzii

Un autoencoder comprimă datele printr-un strat îngust și le reconstruiește

- **Pe scurt:** Dacă rețeaua reconstruiește tranzacțiile normale din cîteva numere, una pe care nu o poate reconstrui este neobișnuită.

$$\mathbf{z} = f_\phi(\mathbf{x}) \in \mathbb{R}^k, \qquad \hat{\mathbf{x}} = g_\psi(\mathbf{z})$$

$$\min_{\phi,\psi} \frac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{x}_i - g_\psi(f_\phi(\mathbf{x}_i)) \right\|^2, \qquad s(\mathbf{x}) = \tfrac{1}{d}\|\mathbf{x} - \hat{\mathbf{x}}\|^2$$

- **unde**
  - f_ϕ, g_ψ = codificatorul și decodorul, perceptroni multistrat mici (MLP) cu unități liniare rectificate (ReLU)
  - k = dimensiunea stratului îngust
  - $s(\mathbf{x})$ = scorul de anomalie
  - Cu f, g liniare se obține subspațiul PCA.

- **Autoencoderul nostru**
  - 29 de intrări (28 de componente anonimizate + logaritmul sumei) → $16 \to 8 \to 16 \to 29$
  - 1 253 de parametri, antrenat numai pe tranzacții legitime

- **Nu are nevoie de etichete**: util pentru fraude noi sau neconfirmate

- **În practică:** un autoencoder modelează "normalul" și semnalează ce diferă, nu neapărat ce este infracțional.

(Bishop & Bishop, 2024, ch. 19)

Exercițiu practic: notebook, secțiunea 1

Experimentul de fraudă: două zile de plăți cu cardul, împărțite în timp

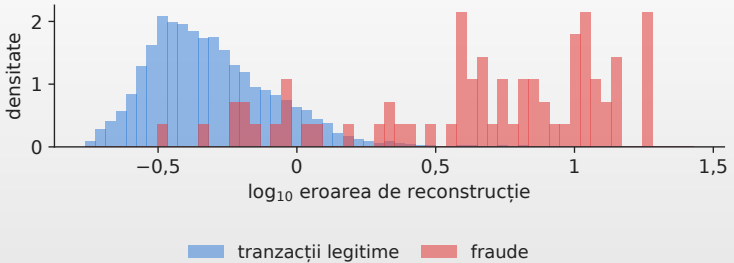
Element	Alegerea
Date	tranzacțiile cu carduri ULB (Dal Pozzolo et al., 2015): 28 924 de rînduri = toate cele 492 de fraude și 10 % dintre cele legitime, reponderate la datele complete
Împărțire în timp	primele 60 % din interval = antrenare (numai legitime, 16 994 de rînduri), următoarele 20 % = validare, ultimele 20 % = test
Test	57 174 de tranzacții din datele complete, 75 de fraude (0,131 %)
Modele de referință	PCA cu 8 componente (autoencoder liniar); regresie logistică antrenată *cu* etichetele de fraudă
Măsuri	aria de sub curba precizie-recall (PR-AUC) și aria de sub curba caracteristicii de operare a receptorului (ROC-AUC), ponderate la datele complete; fraudele printre primele 0,1 % scoruri

- **Fără scurgeri de informație**: mediile și abaterile standard din tranzacțiile legitime de antrenare
  - Oprire timpurie pe partea de validare
  - Semințele 1–3

(Dal Pozzolo et al., 2015)

Exercițiu practic: notebook, secțiunea 1

Eroarea de reconstrucție a plăților frauduloase și a celor legitime

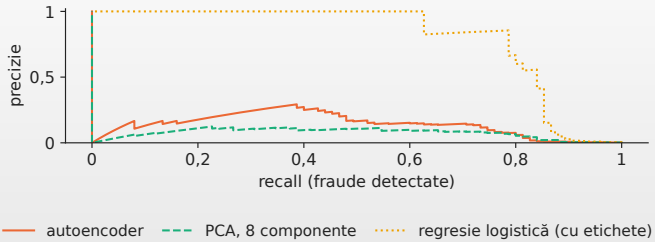


Sursa: ULB Machine Learning Group și Worldline, tranzacții cu carduri (OpenML 1597). Date: estimări proprii; două zile din septembrie 2013; test = ultimele 20 % din interval.

- **Separarea**: eroarea mediană a unei fraude este de 12,7 ori mai mare decît a unei tranzacții legitime (partea de test)
- **Suprapunerea**: multe fraude arată totuși normal pentru autoencoder
  - Se află în grosul distribuției legitime

NN_L8_ae_error_dist

Autoencoderul fără etichete comparat cu un clasificator supervizat

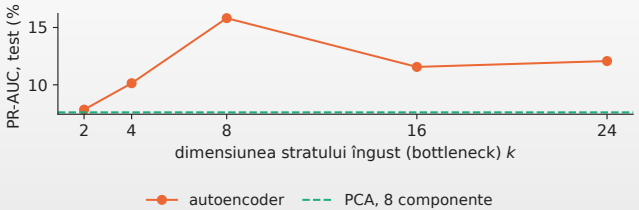


Sursa: ULB Machine Learning Group și Worldline, tranzacții cu carduri (OpenML 1597). Date: estimări proprii; două zile din septembrie 2013; test = ultimele 20 % din interval.

- **PR-AUC pe partea de test**: autoencoder 14,2 % (abatere standard 2,9 pe 3 semințe), PCA 7,6 %, regresie logistică cu etichete 80,4 %
  - Un scor aleator: 0,131 %
- **ROC-AUC** este mare pentru toate trei (94,4, 95,3, 98,2 %): la o fraudă din 700 ascunde diferența pe care o arată PR-AUC

NN_L8_ae_pr

Dimensiunea stratului îngust și detectarea anomaliilor



Sursa: ULB Machine Learning Group și Worldline, tranzacții cu carduri (OpenML 1597). Date: estimări proprii; două zile din septembrie 2013; test = ultimele 20 % din interval.

- **PR-AUC după dimensiunea stratului îngust**: 7,8 % cu $k = 2$, 15,8 % cu $k = 8$, 12,1 % cu $k = 24$ (media pe 2 semințe)
  - Cel mai bun $k = 8$
- **Prea îngust**: chiar și plățile normale sînt reconstruite prost
  - **prea larg**: rețeaua învață să copieze și fraudele

- **În practică:** alegeți k pe o perioadă de validare, niciodată pe fraudele despre care veți raporta.

NN_L8_ae_bottleneck

Fraudele identificate în primele 0,1 % dintre scoruri

Detector	Fraude găsite	Pondere
Autoencoder (fără etichete)	8	10,7 %
PCA, 8 comp. (fără etichete)	0	0,0 %
Regresie logistică (cu etichete)	48	64,0 %

- **Bugetul de alerte**
  - 0,1 % dintre tranzacțiile de test = 57 de alerte de verificat
- **Interpretare**
  - Etichetele valorează mult: învățarea supervizată cîștigă ori de cîte ori există fraude confirmate
  - Autoencoderul este detectorul necunoscutului, un complement, nu un înlocuitor

- **În practică:** folosiți-le pe amîndouă: clasificatorul pentru tiparele cunoscute, autoencoderul pentru a scoate la iveală tipare noi, pe care investigatorii le etichetează.

(calcule proprii, partea de test, ponderat la datele complete)

Exercițiu practic: notebook, secțiunea 1

Cuprins

- Principiile modelării generative
- Autoencodere: detectarea anomaliilor la fraude cu carduri
- Autoencodere variaționale
- Rețele generative adversariale
- Modele de difuzie
- Modelele generative în practică
- Concluzii

VAE transformă autoencoderul într-un generator probabilist

⊡ **Pe scurt:** În locul unui cod pentru fiecare intrare, codificatorul produce o distribuție; decodorul învață să transforme coduri aleatoare în date realiste.

$$\log p_\theta(\mathbf{x}) \;\geq\; \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x} \mid \mathbf{z})\big]}_{\text{reconstrucție}} - \underbrace{\mathrm{KL}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})\big)}_{\text{regularizare}}$$

⊡ **unde**

- Membrul drept = marginea inferioară a verosimilității (ELBO)
- $q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\phi(\mathbf{x}), \mathrm{diag}\,\boldsymbol{\sigma}^2_\phi(\mathbf{x}))$ = codificatorul
- $p_\theta(\mathbf{x} \mid \mathbf{z})$ = decodorul
- $p(\mathbf{z}) = \mathcal{N}(0, I)$ = distribuția a priori
- Al doilea termen este divergența Kullback–Leibler (KL).

⊡ **De ce o margine**

- $\log p(\mathbf{x}) = \mathrm{ELBO} + \mathrm{KL}\big(q(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z} \mid \mathbf{x})\big)$
- Diferența este eroarea codificatorului; derivarea, în Anexa A

⊡ **KL în formă închisă**

- $\mathrm{KL} = \frac{1}{2}\sum_{j=1}^{k}\big(\mu_j^2 + \sigma_j^2 - \log\sigma_j^2 - 1\big)$

⊡ **În practică:** termenul KL face codurile aleatoare să aibă sens: aduce toate intrările în jurul originii spațiului latent.

(Kingma & Welling, 2014; Bishop & Bishop, 2024, sec. 19.2)

Exercițiu practic: notebook, secțiunea 2

Trucul reparametrizării

- **Pe scurt:** Extragerea aleatoare nu este derivabilă, dar o normală standard translatată și scalată este.

$$\mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\sigma}_\phi(\mathbf{x}) \odot \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I)$$
$$\nabla_\phi \, \mathbb{E}_{q_\phi}[h(\mathbf{z})] = \mathbb{E}_\varepsilon \left[\nabla_\phi \, h(\boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \odot \varepsilon) \right]$$

- **unde**
  - $\odot$ = produs element cu element
  - Hazardul este în ε, care nu depinde de ϕ
  - În practică ajunge o extragere a lui ε pentru fiecare imagine și pas.

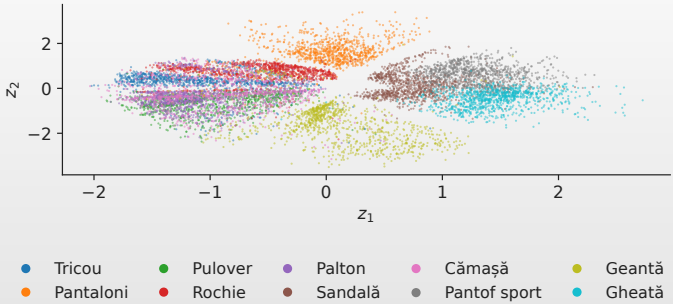
- **VAE-ul nostru pe Fashion-MNIST**
  - Imagini gri 28×28 cu 10 tipuri de îmbrăcăminte, înlocuitorul cifrelor MNIST
  - 60 000 de imagini de antrenare, 10 000 de test; decodor Bernoulli (entropie încrucișată binară)
- **Dimensiunea**
  - $784 \to 400 \to 200 \to k$; 798 816 de parametri pentru $k = 16$; 20 de epoci

- **În practică:** același truc stă la baza oricărui model care extrage aleator în interiorul rețelei: VAE, dropout ca inferență, politici stocastice.

(Kingma & Welling, 2014; Xiao et al., 2017)

Exercițiu practic: notebook, secțiunea 2

Un spațiu latent cu două dimensiuni separă tipurile de produse



Sursa: Fashion-MNIST, Zalando Research (Xiao et al., 2017). Date: estimări proprii; 60.000 de imagini de antrenare și 10.000 de test.

- **Fără etichete**: VAE nu a văzut clasele, dar încălțămintea, gențile și hainele ocupă regiuni separate
- **Măsurat**: un clasificator cu 15 vecini pe cele două medii latente atinge o acuratețe de 71,7 % (5 partiții, imaginile de test)

NN_L8_vae_latent

Calitatea reconstrucțiilor VAE și a eșantioanelor generate

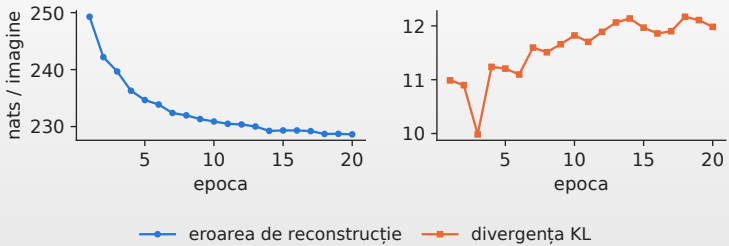


Sursa: Fashion-MNIST, Zalando Research (Xiao et al., 2017). Date: estimări proprii; 60.000 de imagini de antrenare și 10.000 de test.

- **Rîndurile**: imagini reale de test
  - Reconstrucția lor printr-un cod cu 16 dimensiuni
  - Imagini noi decodate din coduri aleatoare $\mathbf{z} \sim \mathcal{N}(0, I)$
- **Neclaritatea** vine din distribuția a posteriori gaussiană și din pierderea calculată pixel cu pixel: decodorul face media detaliilor plauzibile

NN_L8_vae_samples

Compromisul dintre reconstrucție și penalizarea KL în timpul antrenării



Sursa: Fashion-MNIST, Zalando Research (Xiao et al., 2017). Date: estimări proprii; 60.000 de imagini de antrenare și 10.000 de test.

- **Test, $k = 16$, ultima epocă**: reconstrucția 228,6 nats pe imagine, KL 12,0 nats
  - ELBO $-240{,}6$ nats
- **KL crește** cînd eroarea scade: codificatorul folosește mai multă informație pentru fiecare imagine pe măsură ce învață

(calcule proprii, sămînța 1)

NN_L8_vae_elbo

β-VAE: calitatea reconstrucției și structura spațiului latent

$$\mathcal{L}_\beta = -\mathbb{E}_q[\log p(\mathbf{x} \mid \mathbf{z})] + \beta \, \mathrm{KL}\big(q(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})\big)$$

β	Reconstrucție	KL	Dim. active
0,5	222,5	17,8	9
1,0	228,6	12,0	6
4,0	242,5	6,4	4

- β **mai mare**
  - Mai puține dimensiuni latente active (KL pe dimensiune $> 0{,}05$): de la 9 la 4 din 16
  - Reconstrucție mai slabă, coduri mai netede și mai separate
- β **mai mic**
  - Mai aproape de un autoencoder simplu: copii bune, eșantioane slabe

- **În practică:** pentru generarea de scenarii preferați un spațiu latent ordonat ($\beta \geq 1$)
  - Pentru comprimare, reconstrucția ($\beta < 1$).

(Higgins et al., 2017; calcule proprii, imaginile de test, 20 de epoci)

Exercițiu practic: notebook, secțiunea 2

Cuprins

- Principiile modelării generative
- Autoencodere: detectarea anomaliilor la fraude cu carduri
- Autoencodere variaționale
- Rețele generative adversariale
- Modele de difuzie
- Modelele generative în practică
- Concluzii

Rețelele generative adversariale: generatorul și discriminatorul

- **Pe scurt:** Generatorul face date false din zgomot, discriminatorul caută falsul; fiecare progresează împotriva celuilalt.

$$\min_G \max_D \; \mathbb{E}_{\mathbf{x}\sim p_{data}}\Big[\log D(\mathbf{x})\Big] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\Big[\log\Big(1 - D(G(\mathbf{z}))\Big)\Big]$$

$$D^*(\mathbf{x}) = \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_G(\mathbf{x})} \;\Rightarrow\; \text{valoarea} = -\log 4 + 2\,\mathrm{JS}(p_{data} \| p_G)$$

- **unde**
  - G = rețeaua generator, D = discriminatorul (probabilitatea ca $\mathbf{x}$ să fie real)
  - p_G = distribuția lui $G(\mathbf{z})$
  - Divergența Jensen–Shannon (JS), nulă numai cînd $p_G = p_{data}$.

- **La echilibru**
  - $D^* = 1/2$ peste tot: detectivul doar ghicește; pierderea discriminatorului $2\log 2$
- **Fără verosimilitate**
  - Un GAN nu scrie niciodată p_G; doar învață să extragă din ea

- **În practică:** un GAN se judecă după eșantioane (cozi, dependență), nu după verosimilitate.

(Goodfellow et al., 2014; Bishop & Bishop, 2024, ch. 17)

Exercițiu practic: notebook, secțiunea 3

Antrenarea unui GAN: pași alternativi și pierderea nesaturată

- **Pe scurt:** La fiecare pas se actualizează întîi detectivul, apoi falsificatorul.

$$\mathcal{L}_D = -\frac{1}{m}\sum_i \left[\log D(\mathbf{x}_i) + \log(1 - D(G(\mathbf{z}_i)))\right]$$

$$\mathcal{L}_G = -\frac{1}{m}\sum_i \log D(G(\mathbf{z}_i)) \quad \text{(nesaturată)}$$

- **unde**
  - $m = 256$ de zile pe lot
  - $\mathbf{z} \in \mathbb{R}^8$ normal standard
  - $\mathcal{L}_G$ evită gradientul nul al lui $\log(1 - D(G))$ cînd D cîștigă devreme
  - Adam, rata $2 \cdot 10^{-4}$, $\beta_1 = 0{,}5$.

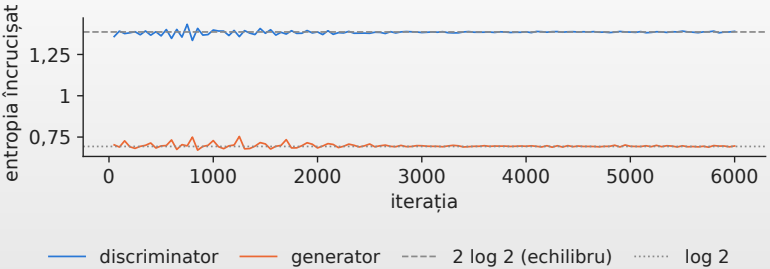
- **GAN-ul nostru**
  - Randamentele logaritmice zilnice ale indicelui S&P 500, 4 ianuarie 2005 – 2019, 3 774 de zile, standardizate
  - Generator cu 4 801 de parametri, discriminator cu 4 353; 6 000 de pași, 3 semințe

- **În practică:** o pierdere a discriminatorului blocată la zero arată că generatorul nu mai învață.

(Goodfellow et al., 2014)

Exercițiu practic: notebook, secțiunea 3

Cele două pierderi se stabilizează lîngă valorile de echilibru

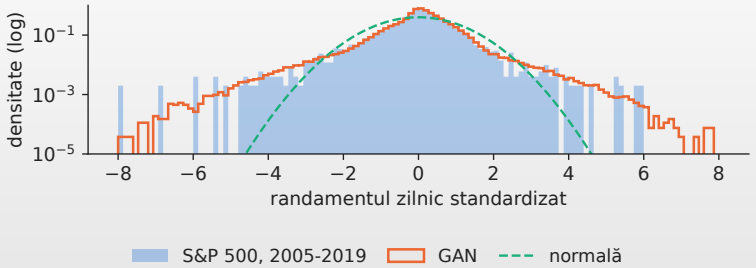


Sursa: Yahoo Finance, cotații zilnice. Date: estimări proprii; S&P 500 4 ianuarie 2005 - 31 decembrie 2019, zile de tranzacționare.

- **Echilibrul**: pierderea discriminatorului oscilează în jurul lui $2\log 2$, iar a generatorului în jurul lui $\log 2$, valorile lui $D^* = 1/2$
- **Nicio garanție de convergență**: jocul pe gradienți poate oscila
  - Curbele de pierdere nu măsoară calitatea eșantioanelor

NN_L8_gan_losses

GAN-ul reproduce cozile groase pe care modelul gaussian nu le surprinde



Sursa: Yahoo Finance, cotații zilnice. Date: estimări proprii; S&P 500 4 ianuarie 2005 - 31 decembrie 2019, zile de tranzacționare.

- **Excesul de aplatizare**: real 11,86, GAN 6,65 (abatere standard 0,48 pe 3 semințe), normala 0
- **Distanța Kolmogorov–Smirnov** față de randamentele reale: GAN 0,030, normala estimată 0,115

(Cont, 2001)

NN_L8_gan_density

Un test de risc: valoarea la risc de 1 % în anii de după antrenare

$$\mathrm{VaR}_{0.01} = -q_{0.01}(r), \qquad \hat{\pi} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{r_t < q_{0.01}\}$$

Metoda (estimată 2005–2019)	Cuantila de 1 %	Depășiri
Simulare istorică	−3,48 %	1,11 %
Eșantioane GAN (3 semințe)	−3,61 %	0,95 %
Normală	−2,67 %	2,39 %

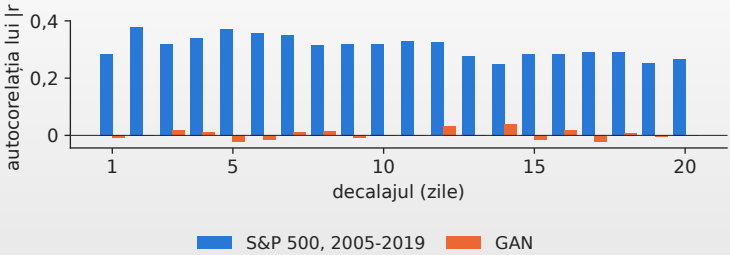
- **Perioada de test**
  - 2 ianuarie 2020 – 30 octombrie 2026, 1 717 de zile; o valoare VaR corectă de 1 % este depășită în circa 17 zile
- **Interpretare**
  - Cuantila GAN este la fel de bună ca simularea istorică; normala este depășită în 2,39 % dintre zile
  - Numai cuantile necondiționate: niciun model de aici nu reacționează la volatilitate
  - Pierderea așteptată în coadă (ES), media pierderilor dincolo de VaR, depinde de aceeași coadă

- **În practică:** un generator este atît de bun cît măsura de risc pe care o reproduce: testați-l retrospectiv ca pe orice model VaR (Kupiec, 1995).

(calcule proprii)

Exercițiu practic: notebook, secțiunea 3

Un GAN pe zile izolate nu poate învăța gruparea volatilității



Sursa: Yahoo Finance, cotații zilnice. Date: estimări proprii; S&P 500 4 ianuarie 2005 - 31 decembrie 2019, zile de tranzacționare.

- **Randamentele reale**: autocorelația lui $|r_t|$ este 0,28 la decalajul 1 și încă 0,27 la decalajul 20: perioadele calme și cele agitate se grupează
- **Eșantioanele GAN** sînt extrageri independente ($-0{,}01$ la decalajul 1): distribuția marginală corectă, dinamica greșită
  - Modelele de secvențe (TimeGAN, Quant GANs) generează traiectorii întregi

(Cont, 2001; Yoon et al., 2019; Wiese et al., 2020)

NN_L8_gan_acf

Colapsul modurilor și GAN-ul Wasserstein

- **Pe scurt:** Un generator poate cîștiga producînd cîteva eșantioane convingătoare; distanța Wasserstein dă gradienți utili chiar cînd suporturile nu se suprapun.

$$W_1(p_{data}, p_G) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{p_{data}}[f(\mathbf{x})] - \mathbb{E}_{p_G}[f(\mathbf{x})]$$

- **unde**
  - f = rețeaua critic, constrînsă să fie 1-Lipschitz (tăierea ponderilor sau o penalizare a gradientului)
  - Generatorul minimizează W_1: GAN-ul Wasserstein (WGAN).

- **Colapsul modurilor**
  - Generatorul duce toți $\mathbf{z}$ în aceeași regiune: de exemplu, numai zile calme, niciodată prăbușiri
- **Verificări**
  - Comparați cuantilele din coadă, ponderea zilelor extreme și dispersia eșantioanelor cu datele reale

- **În practică:** un generator pentru teste de stres care a colapsat pe zilele calme este mai rău decît lipsa unui generator.

(Arjovsky, Chintala & Bottou, 2017)

Cuprins

- Principiile modelării generative
- Autoencodere: detectarea anomaliilor la fraude cu carduri
- Autoencodere variaționale
- Rețele generative adversariale
- Modele de difuzie
- Modelele generative în practică
- Concluzii

Modelele de difuzie: adăugarea și eliminarea treptată a zgomotului

- **Pe scurt:** Cu zgomot gaussian adăugat treptat, orice date devin zgomot pur; o rețea învață fiecare pas înapoi.

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\ \beta_t I\right)$$

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\ (1-\bar{\alpha}_t)I\right), \quad \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$$

- **unde**
  - $\mathbf{x}_0$ = o observație reală (aici o curbă a randamentelor standardizată)
  - β_t = programul zgomotului, liniar de la 10^{-4} la 0,02 în $T = 200$ de pași
  - $\bar{\alpha}_T \approx 0$, deci $\mathbf{x}_T$ este aproape normal standard.

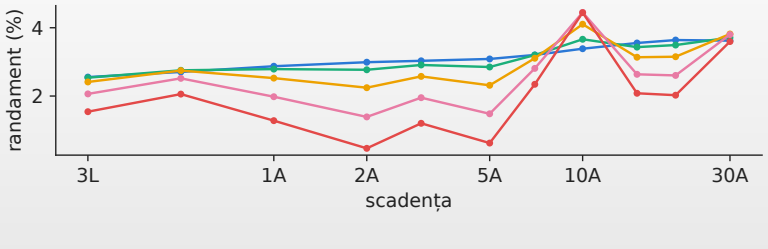
- **Formă închisă**
  - Orice pas t se obține dintr-o singură extragere: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon$
- **Fără parametri**
  - Procesul înainte este fix; numai procesul invers se învață

- **În practică:** difuzia transformă generarea în multe probleme mici de regresie
  - De aceea se antrenează stabil.

(Sohl-Dickstein et al., 2015; Ho, Jain & Abbeel, 2020)

Exercițiu practic: notebook, secțiunea 4

Procesul înainte pe o curbă reală a randamentelor



Sursa: Banca Centrală Europeană, curba randamentelor AAA din zona euro. Date: curba din 30 octombrie 2026; programul de zgomot din curs (mecanism).

- **Curba din 30 octombrie 2026**, 11 scadențe, de la 3 luni la 30 de ani
  - La $t = 199$ forma a dispărut și a rămas doar zgomot
- **Rețeaua** trebuie să învețe drumul înapoi, de la dreapta la stînga, cîte un pas mic

(mecanism ilustrat pe date reale; o singură extragere a zgomotului)

NN_L8_diffusion_forward

Învățarea eliminării zgomotului: prezicem zgomotul adăugat

- **Pe scurt:** Rețeaua vede curba cu zgomot și pasul t și ghicește zgomotul; scăzîndu-l pas cu pas, obținem o curbă nouă.

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\varepsilon}}\left\| \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\boldsymbol{\theta}}\left(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\varepsilon},\ t\right)\right\|^2$$

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right) + \sqrt{\beta_t}\,\mathbf{u}, \quad \mathbf{u} \sim \mathcal{N}(0, I)$$

- **unde**
  - $\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}$ = rețeaua care elimină zgomotul
  - $\alpha_t = 1 - \beta_t$
  - t uniform pe $\{1, \ldots, T\}$
  - Pierderea este un ELBO simplificat și reponderat.

- **Rețeaua noastră**
  - MLP cu 3 straturi ascunse de 256 și o reprezentare sinusoidală a lui t: 145 675 de parametri
  - 12 000 de pași cu loturi de 512 curbe; 5.000 de curbe generate pe sămînță

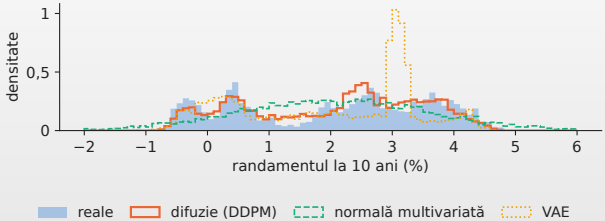
- **Date**
  - 5 664 de curbe zilnice ale Băncii Centrale Europene, 6 septembrie 2004 – 30 octombrie 2026

- **În practică:** aceeași pierdere stă la baza generatoarelor de imagini, iar aici generează scenarii de dobîndă.

(Ho, Jain & Abbeel, 2020; Bishop & Bishop, 2024, ch. 20)

Exercițiu practic: notebook, secțiunea 4

Cele două regimuri ale dobînzilor euro în eșantioanele generate prin difuzie și de modelul gaussian

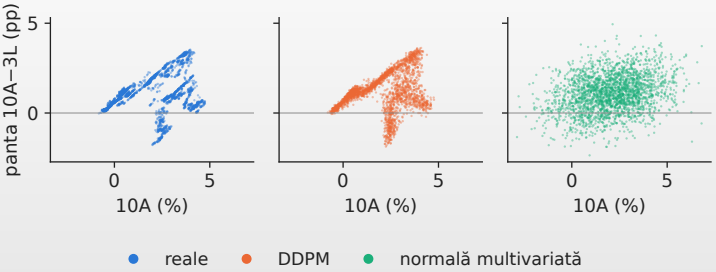


Sursa: Banca Centrală Europeană, curba randamentelor AAA din zona euro. Date: estimări proprii; 6 septembrie 2004 - 30 octombrie 2026, zile lucrătoare.

- **Randamentul real la 10 ani**: de la −0,82 % la 4,78 %, negativ în 13,8 % dintre zile: un regim de dobînzi scăzute și unul de dobînzi normale
- **Distanța Wasserstein** dintre randamentul generat la 10 ani și cel real: DDPM 0,149, VAE 0,191, normala 0,325 puncte procentuale

NN_L8_yc_density

Difuzia păstrează dependența neliniară dintre nivel și pantă

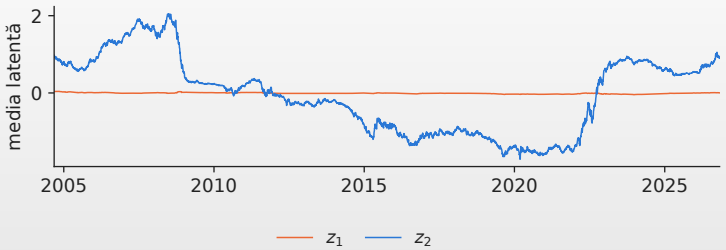


Sursa: Banca Centrală Europeană, curba randamentelor AAA din zona euro. Date: estimări proprii; 6 septembrie 2004 - 30 octombrie 2026, zile lucrătoare.

- **Curbe inversate** (10 ani sub 3 luni): 8,7 % dintre zilele reale, 10,2 % dintre eșantioanele difuziei, 14,7 % dintre cele normale
- **Distanța pantei** (Wasserstein): DDPM 0,099, VAE 0,242, normala 0,113

NN_L8_yc_scatter

Un VAE cu doi factori ai curbei redescoperă nivelul și panta



Sursa: Banca Centrală Europeană, curba randamentelor AAA din zona euro. Date: estimări proprii; 6 septembrie 2004 - 30 octombrie 2026, zile lucrătoare.

- **Corelațiile** mediilor latente cu nivelul (media celor 11 randamente) și cu panta (10 ani minus 3 luni): pînă la 0,99 și 0,66 în valoare absolută
- **Comprimare interpretabilă**: traiectoria latentă urmărește perioada de dobînzi scăzute 2014–2021 și înăsprirea din 2022, fără să i se spună

(Svensson, 1994)

NN_L8_yc_latent_time

Cuprins

- Principiile modelării generative
- Autoencodere: detectarea anomaliilor la fraude cu carduri
- Autoencodere variaționale
- Rețele generative adversariale
- Modele de difuzie
- Modelele generative în practică
- Concluzii

Criterii de evaluare a datelor financiare sintetice

- **Fidelitatea**
  - Distribuțiile marginale și cozile (aplatizarea, cuantila de 1 %), dependența între active și în timp (autocorelația lui $|r_t|$)
- **Diversitatea**
  - Fără colaps al modurilor: ponderea zilelor extreme și a regimurilor ca în date
- **Utilitatea**
  - Antrenare pe date sintetice, test pe date reale (TSTR): un model estimat pe date sintetice trebuie să funcționeze pe date reale
- **Confidențialitatea**
  - Generatoarele pot memora înregistrări: verificați distanța fiecărui eșantion pînă la cea mai apropiată înregistrare reală
- **Fișa noastră de evaluare**
  - Randamentele GAN: cozile corecte, gruparea volatilității greșită
  - Curbele DDPM: regimurile și inversările corecte, normala greșită

- **În practică:** nu folosiți date sintetice într-o decizie înainte ca ele să treacă aceleași teste retrospective ca modelul pe care îl alimentează.

(Cont, 2001; Yoon et al., 2019)

Alegerea unui model generativ

Nevoia	AE	VAE	GAN	DDPM
Scor de anomalie	da (eroarea)	da (ELBO)	indirect	da (marginea)
Calitatea eșantioanelor	–	neclare	clare	cea mai bună
Viteza generării	–	o trecere	o trecere	200 de treceri
Stabilitatea antrenării	mare	mare	mică	mare
Factori latenți	da	da, netezi	fără invers	nu

- **Regulă practică**: autoencodere pentru monitorizare, VAE pentru comprimare și explorare, difuzie pentru simulări fidele, GAN cînd generarea trebuie să fie rapidă

(Bishop & Bishop, 2024, ch. 16–20)

Cuprins

- Principiile modelării generative
- Autoencodere: detectarea anomaliilor la fraude cu carduri
- Autoencodere variaționale
- Rețele generative adversariale
- Modele de difuzie
- Modelele generative în practică
- Concluzii

Idei principale

- **Autoencoderele** modelează "normalul": la fraudele cu carduri, PR-AUC 14,2 % fără etichete, față de 80,4 % pentru un clasificator cu etichete
- **VAE** maximizează ELBO = reconstrucție − KL
  - Trucul reparametrizării le face antrenabile
  - β schimbă detaliul pe un spațiu latent ordonat
- **GAN** reproduc cozile groase (aplatizare 6,65, față de 11,86 real) și o valoare VaR de 1 % utilizabilă, dar nu și gruparea volatilității cînd sînt antrenate pe zile izolate
- **Difuzia** învață să elimine zgomotul
  - Pe curbele euro păstrează ambele regimuri de dobîndă și inversările pe care distribuția normală nu le vede

Cercetările grupului nostru: modele generative pentru riscul extrem

- **Modele mari de limbaj (LLM) ca instrumente de prognoză a riscului: LLM-VaR și LLM-ES** (Pele, Bolovăneanu, Lin, Ren, Ginavar, Spilak, Andrei, Toma, Lessmann & Härdle, 2026)
  - Lucrarea folosește modele de limbaj generative, cărora li se dau istoricele randamentelor, pentru prognoza valorii la risc și a pierderii așteptate în coadă, evaluate cu testele retrospective standard
- **Modele fundaționale pentru riscul extrem** (Jheng, Tak, Găman, Mazurencu-Marinescu-Pele & Pele, 2026)
  - Lucrarea folosește modele probabiliste de prognoză preantrenate, care generează traiectorii, fără reantrenare și cu calibrare conformală, pentru prognoza VaR și ES pe piețele din Europa Centrală și de Est
- **Legătura cu acest curs**: ambele lucrări tratează un generator de scenarii ca pe un model de risc și îl judecă prin testele retrospective folosite aici pentru GAN

Rezultatele de pe acest slide sînt cele din lucrări; cifrele de pe celelalte slide-uri provin din calculele noastre.

(Academia de Studii Economice din București; Academia Română, Institutul de Prognoză Economică)

Lecturi și resurse suplimentare

- **Bishop & Bishop (2024)**, capitolele 16–20: variabile latente, GAN, fluxuri, autoencodere, difuzie
- **Notele lui Lilian Weng** despre modelele de difuzie: lilianweng.github.io.
- **Ho et al. (2020)**, lucrarea DDPM, scurtă și ușor de citit
- **Acest curs**: cod, date, notebook-uri — github.com/danpele/NEURAL_BIZ.
- **Exercițiu practic**: notebook-ul reproduce toate graficele și cifrele acestui curs

Test 1: autoencoder

- De ce se antrenează autoencoderul de fraudă numai pe tranzacții legitime?

Răspuns

- Trebuie să învețe cum arată "normalul"
  - Fraudele sînt apoi reconstruite prost și primesc un scor mare (eroarea mediană de 12,7 ori mai mare)

Test 2: ELBO

- Care sînt cei doi termeni ai ELBO și ce face fiecare?

Răspuns

- Log-verosimilitatea așteptată a datelor dat fiind codul (reconstrucția) minus divergența KL a codificatorului față de distribuția a priori (ține codurile lîngă $\mathcal{N}(0, I)$, ca un cod aleator să producă date realiste)

Test 3: echilibrul GAN

- Ce produce discriminatorul optim cînd generatorul reproduce datele?

Răspuns

- $D^*(\mathbf{x}) = 1/2$ peste tot, pentru că $p_{data} = p_G$
  - Pierderea lui este atunci $2 \log 2$

Test 4: randamente sintetice

- GAN-ul nostru reproduce corect cozile randamentelor S&P 500.
  - De ce nu este bun pentru un model de volatilitate?

Răspuns

- Extrage zilele independent: autocorelația lui $|r_t|$ este $-0{,}01$ în loc de $0{,}28$
  - Gruparea volatilității cere un generator de secvențe

Test 5: difuzia

- Ce prezice rețeaua de difuzie în timpul antrenării?

Răspuns

- Zgomotul gaussian ε adăugat lui $\mathbf{x}_0$ pentru a obține $\mathbf{x}_t$, dați fiind $\mathbf{x}_t$ și t
  - Generarea scade pas cu pas zgomotul prezis

Bibliografie (1/2)

- Arjovsky, M., Chintala, S. & Bottou, L. (2017). Wasserstein generative adversarial networks. *Proceedings of ICML*, PMLR 70, 214–223. arXiv:1701.07875
- Bishop, C. M. & Bishop, H. (2024). *Deep Learning: Foundations and Concepts*, ch. 16–20. Springer. doi:10.1007/978-3-031-45468-4
- Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2), 223–236. doi:10.1080/713665670
- Dal Pozzolo, A., Caelen, O., Johnson, R. A. & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. *IEEE Symposium Series on Computational Intelligence*, 159–166. doi:10.1109/SSCI.2015.33
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27. arXiv:1406.2661
- Higgins, I., Matthey, L., Pal, A. et al. (2017). beta-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*. openreview.net
- Ho, J., Jain, A. & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33. arXiv:2006.11239
- Jheng, S.-L., Tak, R., Găman, Ș., Mazurencu-Marinescu-Pele, M. & Pele, D. T. (2026). Can foundation models manage risk? Zero-shot VaR and ES forecasting with conformal calibration in CEE markets. *Economic Computation and Economic Cybernetics Studies and Research*, 60(2), 267–290. doi:10.24818/18423264/60.2.26.14
- Kingma, D. P. & Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations*. arXiv:1312.6114

Bibliografie (2/2)

- Kupiec, P. H. (1995). Techniques for verifying the accuracy of risk measurement models. *Journal of Derivatives*, 3(2), 73–84. doi:10.3905/jod.1995.407942
- Pele, D. T., Bolovăneanu, V., Lin, M.-B., Ren, R., Ginavar, A. T., Spilak, B., Andrei, A.-V., Toma, F.-M., Lessmann, S. & Härdle, W. K. (2026). In the beginning was the word: LLM-VaR and LLM-ES. SSRN Working Paper 5104383. doi:10.2139/ssrn.5104383
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *Proceedings of ICML*, PMLR 37, 2256–2265. arXiv:1503.03585
- Svensson, L. E. O. (1994). Estimating and interpreting forward interest rates: Sweden 1992–1994. NBER Working Paper 4871. doi:10.3386/w4871
- Wiese, M., Knobloch, R., Korn, R. & Kretschmer, P. (2020). Quant GANs: Deep generation of financial time series. *Quantitative Finance*, 20(9), 1419–1440. doi:10.1080/14697688.2020.1730426
- Xiao, H., Rasul, K. & Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747
- Yoon, J., Jarrett, D. & van der Schaar, M. (2019). Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32. papers.nips.cc/paper/8789

Anexa A: derivarea ELBO

$$\log p(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z}\mid\mathbf{x})}\right] + \mathbb{E}_q\left[\log \frac{q(\mathbf{z}\mid\mathbf{x})}{p(\mathbf{z}\mid\mathbf{x})}\right]$$

$$= \underbrace{\mathbb{E}_q[\log p(\mathbf{x}\mid\mathbf{z})] - \mathrm{KL}(q\|p(\mathbf{z}))}_{\text{ELBO}} + \underbrace{\mathrm{KL}(q\|p(\mathbf{z}\mid\mathbf{x}))}_{\geq 0}$$

- Se înmulțește și se împarte cu $q(\mathbf{z}\mid\mathbf{x})$ în $\log p(\mathbf{x}) = \log p(\mathbf{x},\mathbf{z}) - \log p(\mathbf{z}\mid\mathbf{x})$ și se ia media după q
- A doua divergență KL este nenegativă, deci ELBO este o margine inferioară
  - Este exactă cînd q este egală cu distribuția a posteriori adevărată
- Pentru gaussienele $\mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}\,\boldsymbol{\sigma}^2)$ și $\mathcal{N}(0, I)$, divergența KL este $\frac{1}{2}\sum_j(\mu_j^2 + \sigma_j^2 - \log\sigma_j^2 - 1)$

(Kingma & Welling, 2014)

Anexa B: discriminatorul optim

$$V(D) = \int \left[p_{data}(\mathbf{x}) \log D(\mathbf{x}) + p_G(\mathbf{x}) \log(1 - D(\mathbf{x})) \right] d\mathbf{x}$$

$$\frac{\partial}{\partial D} : \ \frac{p_{data}}{D} - \frac{p_G}{1 - D} = 0 \ \Rightarrow \ D^* = \frac{p_{data}}{p_{data} + p_G}$$

- Se maximizează integrandul punct cu punct: $a \log y + b \log(1 - y)$ are maximul în $y = a/(a + b)$
- Înlocuind D^* se obține $V = -\log 4 + \mathrm{KL}(p_{data} \| m) + \mathrm{KL}(p_G \| m)$, $m = (p_{data} + p_G)/2$, adică $-\log 4 + 2\,\mathrm{JS}$

(Goodfellow et al., 2014)

Anexa C: setările celor patru experimente

Model	Setări
AE	29–16–8–16–29, ReLU, Adam 10^{-3}, lot 256, cel mult 60 de epoci, răbdare 6 pe eroarea de validare; semințele 1–3
VAE	784–400–200–k, $k \in \{2, 16\}$, Adam 10^{-3}, lot 128, 20 de epoci, decodor Bernoulli
GAN	zgomot 8; două straturi ascunse de 64, LeakyReLU 0,2; Adam $2 \cdot 10^{-4}$, $\beta_1 = 0{,}5$; 6.000 de pași cu loturi de 256 de zile
DDPM	$T = 200$, β_t liniar; MLP 3×256 cu reprezentarea timpului; Adam 10^{-3}, scădere cosinus; 12.000 de pași cu loturi de 512
Echipament	laptop cu procesor Apple; rularea completă 0,3 minute