

Rețele neuronale și deep learning

Cursul 7: Atenția și transformerele



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

- De ce atenția a înlocuit recurența pentru text
  - Un singur pas între oricare două cuvinte, calculat în paralel
- Formulele atenției cu produs scalar scalat, ale atenției cu mai multe capete, ale codificării poziției și ale blocului transformer, cu toate simbolurile definite
- Compararea, pe tweet-uri financiare reale, a unui transformer mic antrenat de la zero, a unui model Bidirectional Encoder Representations from Transformers (BERT) preantrenat pentru finanțe (FinBERT) și a unui model de referință bazat pe frecvența termenului și inversul frecvenței în documente (TF-IDF)
- Citirea atentă a hărților de atenție
  - Cînd preantrenarea, ajustarea fină sau un model simplu sînt alegerea potrivită pentru o firmă

Abrevieri

ASE	Academia de Studii Economice din București
BCE	Banca Centrală Europeană
BERT	Bidirectional Encoder Representations from Transformers (reprezentări bidirecționale ale codificatorului transformer)
CNN	rețea neuronală convoluțională
ES	pierderea așteptată în coadă (expected shortfall)
FFN	rețea directă aplicată fiecărei poziții (feed-forward)
FinBERT	BERT antrenat suplimentar pe texte financiare
GELU	unitate liniară cu eroare gaussiană (Gaussian error linear unit)
GPT	Generative Pre-trained Transformer (transformer generativ preantrenat)
IPE	Institutul de Prognoză Economică, Academia Română
LLM	model mare de limbaj
LN	normalizarea pe strat (layer normalisation)
LSTM	rețea cu memorie pe termen lung și scurt (long short-term memory)
MHA	atenția cu mai multe capete (multi-head attention)
MLM	modelarea limbajului cu cuvinte mascate (masked language modelling)
MPS	Metal Performance Shaders (interfața procesorului grafic Apple)
NLP	prelucrarea limbajului natural
ReLU	unitate liniară rectificată (rectified linear unit)
RNN	rețea neuronală recurentă
TF-IDF	frecvența termenului–inversul frecvenței în documente (TF-IDF)
VaR	valoarea la risc (Value-at-Risk)

Cuprins

- Motivația mecanismului de atenție
- Mecanismul atenției
- Arhitectura transformer
- Studiu de caz: sentimentul tweet-urilor financiare
- Transformerele în practică
- Concluzii

Știrile financiare ca date de intrare pentru modele

- **Pe scurt:** Mii de titluri și postări sosesc în fiecare oră; o mesă de tranzacționare are nevoie de tonul lor în cîteva secunde, consecvent, pe aceeași scară.

$$\hat{\mathbf{p}}(x_{1:n}) = \text{softmax}\left(W_o\,\mathbf{h}(x_{1:n}) + \mathbf{b}_o\right) \in \Delta^3$$

- **unde**
  - $x_{1:n}$ = tokenii unui text
  - $\mathbf{h}$ = o reprezentare de dimensiune fixă a textului
  - Δ^3 = probabilitățile pentru negativ (bearish), pozitiv (bullish), neutru
  - Tot cursul este despre cum se calculează $\mathbf{h}$.

- **Datele noastre**
  - 9 543 de tweet-uri financiare etichetate pentru antrenare, 2 388 pentru test
  - Test: 14,5 % negative, 19,9 % pozitive, 65,6 % neutre
- **Dovezi**
  - Pesimismul din presă anticipează randamentele și volumul din ziua următoare (Tetlock, 2007)

- **În practică:** un model de sentiment este un instrument de măsură: trebuie să fie precis pe știrile negative rare, nu doar în medie.

(Tetlock, 2007; Malo et al., 2014)

Limitele rețelelor recurente pentru textele lungi

- **Pe scurt:** O rețea recurentă comprimă tot ce a citit pînă acum într-un singur vector de stare, actualizat token cu token.

$$\mathbf{h}_t = \phi\big(W_h\mathbf{h}_{t-1} + W_x\mathbf{x}_t + \mathbf{b}\big), \qquad \frac{\partial \mathbf{h}_n}{\partial \mathbf{h}_1} = \prod_{t=2}^{n} \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}}$$

- **unde**
  - $\mathbf{x}_t$ = reprezentarea tokenului t
  - $\mathbf{h}_t$ = starea
  - ϕ = tanh
  - Produsul a $n-1$ matrice jacobiene scade sau explodează odată cu distanța dintre cuvinte.

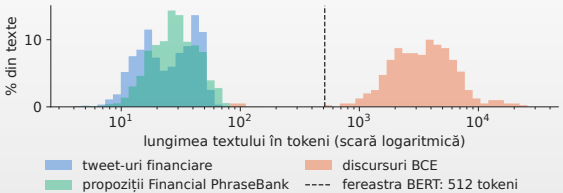
- **Două costuri**
  - Drumul dintre cuvintele i și j: $|i-j|$ pași, deci legăturile lungi se estompează
  - n pași succesivi: nicio paralelizare pe propoziție

- **Porțile ajută, dar nu vindecă**
  - Rețeaua cu memorie pe termen lung și scurt (LSTM) din cursul 5 atenuează, dar păstrează drumul secvențial

- **În practică:** în "profitul a scăzut, dar prognoza a fost majorată", verdictul depinde de un cuvînt aflat departe de subiect.

(Bahdanau, Cho & Bengio, 2015; Bishop & Bishop, 2024, sec. 12.1)

Lungimea tweet-urilor și a discursurilor băncilor centrale, în tokeni



Sursa: Twitter financial news (zeroshot); Malo et al. (2014); discursurile Băncii Centrale Europene; tokenizatorul modelului ProsusAI. Date: tweet-uri financiare; Financial PhraseBank; discursurile Băncii Centrale Europene 8 ianuarie 2022 - 30 octombrie 2026.

- **Lungimea mediană în tokeni**: tweet-uri 25, propoziții Financial PhraseBank 28, discursuri ale Băncii Centrale Europene (BCE) 3 338
- **Limita de context**: 98,8 % dintre cele 400 de discursuri depășesc 512 tokeni
  - Doar 0,19 % dintre tweet-uri depășesc 64

- **În practică:** potriviți modelul cu lungimea textului: un tweet începe în orice transformer, un discurs cere împărțire sau un model cu context lung.

NN_L7_context_lengths

Cuprins

- Motivația mecanismului de atenție
- Mecanismul atenției
- Arhitectura transformer
- Studiu de caz: sentimentul tweet-urilor financiare
- Transformerele în practică
- Concluzii

Atenția este o căutare aproximativă într-un dicționar

- **Pe scurt:** Fiecare cuvînt pune o întrebare (query), fiecare cuvînt oferă o etichetă (key) și un conținut (value); răspunsul este media valorilor, ponderată după potrivirea cheilor cu întrebarea.

$$\mathbf{y}_i = \sum_{j=1}^{n} a_{ij}\,\mathbf{v}_j, \qquad a_{ij} = \frac{\exp\left(\mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d_k}\right)}{\sum_{l=1}^{n} \exp\left(\mathbf{q}_i^\top \mathbf{k}_l / \sqrt{d_k}\right)}$$

$$\mathbf{q}_i = W_Q^\top \mathbf{x}_i, \quad \mathbf{k}_j = W_K^\top \mathbf{x}_j, \quad \mathbf{v}_j = W_V^\top \mathbf{x}_j$$

- **unde**
  - $\mathbf{x}_i \in \mathbb{R}^d$ = reprezentarea tokenului
  - $W_Q, W_K \in \mathbb{R}^{d \times d_k}$, $W_V \in \mathbb{R}^{d \times d_v}$ = proiecții învățate
  - $a_{ij} \geq 0$, $\sum_j a_{ij} = 1$: o distribuție de probabilitate pe cuvinte.

- **Căutarea exactă**
  - Un dicționar întoarce valoarea singurei chei potrivite
- **Căutarea aproximativă**
  - Atenția întoarce un amestec al tuturor valorilor: derivabil, deci W_Q, W_K, W_V se învață
  - Un singur pas între oricare două cuvinte, oricare ar fi distanța

- **În practică:** fiecare cuvînt de ieșire este un amestec care ține cont de context: "prognoza" poate împrumuta sens de la "majorată", aflat zece cuvinte mai departe.

(Vaswani et al., 2017; Bishop & Bishop, 2024, sec. 12.1)

Exercițiu practic: notebook, secțiunea 2

Atenția în formă matriceală, pentru toate cuvintele simultan

- **Pe scurt:** Așezînd întrebările, cheile și valorile în matrice, atenția devine două produse de matrice și un softmax pe rînduri.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + \mathbf{M}\right)V$$
$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

- **unde**
  - $X \in \mathbb{R}^{n \times d}$ = cele n reprezentări ale tokenilor
  - $QK^\top \in \mathbb{R}^{n \times n}$ = toate scorurile perechilor
  - $\mathbf{M}$ = masca, 0 pentru perechile permise și $-\infty$ pentru completare (sau pentru tokenii viitori, într-un decodor).

- **Costul pe strat**
  - $O(n^2 d)$ operații și memorie $O(n^2)$
  - Un discurs median costă de circa 17 828 de ori mai mult decît un tweet median
- **Paralel**
  - Fără buclă pe poziții: procesoarele grafice înmulțesc toată propoziția deodată

- **În practică:** termenul n^2 explică de ce ferestrele de context sînt limitate și rapoartele lungi trebuie tăiate în bucăți.

(Vaswani et al., 2017)

Exercițiu practic: notebook, secțiunea 2

Scalarea cu $\sqrt{d_k}$ și saturația softmax-ului

- **Pe scurt:** Produsele scalare ale unor vectori aleatori lungi cresc odată cu lungimea lor; fără scalare, softmax-ul devine one-hot și gradientul dispare.

$$q_l, k_l \overset{\text{i.i.d.}}{\sim} (0,1) \;\Rightarrow\; \mathbb{E}[\mathbf{q}^\top \mathbf{k}] = 0, \quad \mathrm{Var}(\mathbf{q}^\top \mathbf{k}) = \sum_{l=1}^{d_k} \mathrm{Var}(q_l k_l) = d_k$$

$$\mathrm{Var}\left(\mathbf{q}^\top \mathbf{k} / \sqrt{d_k}\right) = 1 \quad \text{pentru orice } d_k$$

- **unde**
  - d_k = dimensiunea întrebărilor și cheilor (64 pe cap în BERT)
  - Cu scoruri de variață unu, softmax-ul își împarte masa, iar jacobianul $\mathrm{diag}(\mathbf{a}) - \mathbf{a}\mathbf{a}^\top$ nu se apropie de zero.

- **Fără scalare**
  - Cu $d_k = 64$ scorurile au abaterea standard 8, deci un singur cuvînt ia aproape toată ponderea
- **Interpretare ca temperatură**
  - $\sqrt{d_k}$ este o temperatură fixă care păstrează atenția difuză la inițializare

- **În practică:** o constantă de proiectare, nu un parametru de calibrat: datorită ei se pot antrena stive adînci de straturi de atenție.

(Vaswani et al., 2017)

Atenția cu mai multe capete: mai multe relații simultan

- **Pe scurt:** Un cap mediază un singur tip de relație; mai multe capete, cu proiecții mai mici, surprind în paralel sintaxa, negația și entitățile.

$$\mathrm{MHA}(X) = \left[\mathbf{H}_1 \,\|\, \cdots \,\|\, \mathbf{H}_h\right] W_O$$
$$\mathbf{H}_m = \mathrm{Attention}(XW_Q^m, XW_K^m, XW_V^m)$$

- **unde**
  - h = numărul de capete
  - $W_Q^m, W_K^m, W_V^m \in \mathbb{R}^{d \times d/h}$
  - $\|$ = concatenare
  - $W_O \in \mathbb{R}^{d \times d}$
  - Parametri: $4d^2$ (plus termenii liberi), ca la un singur cap mare.

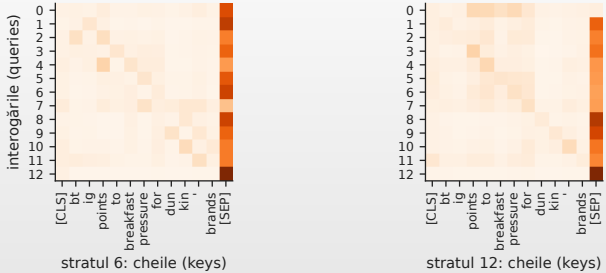
- **BERT-base**
  - 12 straturi $\times$ 12 capete, $d = 768$, $d/h = 64$
- **Transformerul nostru mic**
  - 2 straturi $\times$ 4 capete, $d = 64$, $d/h = 16$

- **În practică:** capetele sînt specialiști ieftini: același cost ca un singur cap, mai multe tipuri de context pe strat.

(Vaswani et al., 2017; Clark et al., 2019)

Exercițiu practic: notebook, secțiunea 2

La ce este atent FinBERT într-un tweet negativ real



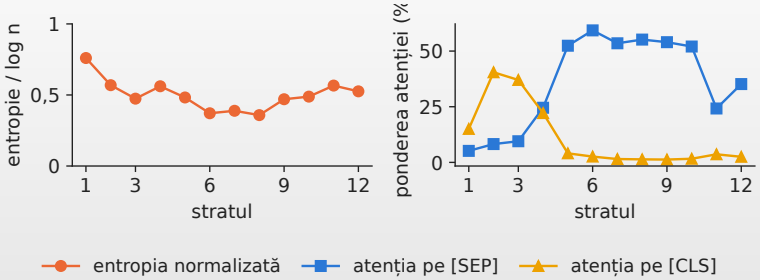
Sursa: Seturile Twitter financial news (Hugging Face, zeroshot). Date: un tweet de test; FinBERT preantrenat, atenția mediată pe 12 capete.

- **Tweet-ul** (setul de test, etichetă negativă): "BTIG points to breakfast pressure for Dunkin' Brands"
  - FinBERT preantrenat îi dă doar 7,3 % probabilitate de a fi negativ: tonul negativ este implicit
- **Stratul 6**: tokenul [CLS] pune 5,7 % din atenție pe "points"
  - În stratul 12, cea mai mare parte a ponderii merge spre tokenii separatori

(calcule proprii; media pe cele 12 capete)

NN_L7_attention_example

De la un strat la altul, atenția se concentrează pe tokenii separatori



Sursa: Seturile Twitter financial news (Hugging Face, zeroshot). Date: 500 de tweet-uri de test; FinBERT preantrenat.

- **Entropia** rîndurilor de atenție, împărțită la $\log n$: 0,76 în stratul 1, minimul 0,36 în stratul 8 (1 = uniformă)
- **[SEP] ca destinație implicită**: pînă la 59,3 % din toată atenția (stratul 6) merge spre separator, o operație nulă pe care un cap o folosește cînd nu are nimic de adăugat

(calcule proprii, 500 de tweet-uri de test; Clark et al., 2019)

NN_L7_attention_entropy

Ponderile de atenție ca explicații: limite

- **Pe scurt:** Unde se uită un cap nu este același lucru cu motivul deciziei: valorile, straturile următoare și clasificatorul transformă tot ce selectează atenția.

$$\mathbf{h}_i^{(l+1)} = \mathbf{h}_i^{(l)} + \sum_{m=1}^{h} \sum_j a_{ij}^{(l,m)} W_O^{m\top} W_V^{m\top} \mathbf{h}_j^{(l)} + \dots$$

- **unde**
  - Contribuția cuvîntului j este a_{ij} înmulțit cu un vector de valori a cărui normă poate fi mare sau mică
  - Fluxul rezidual $\mathbf{h}_i^{(l)}$ duce informație pe care atenția nu o atinge.

- **Dovezi**
  - Hărți de atenție diferite pot da aceeași predicție (Jain & Wallace, 2019)
  - Atenția mare pe [SEP] nu poartă aproape nicio informație (Clark et al., 2019)
- **Instrumente mai bune**
  - Atribuiri bazate pe gradient și teste de ocluzie (cursul 11)

- **În practică:** nu prezentați niciodată unui supraveghetor o hartă de atenție drept "motivul" unei decizii.

(Jain & Wallace, 2019; Clark et al., 2019)

Cuprins

- Motivația mecanismului de atenție
- Mecanismul atenției
- Arhitectura transformer
- Studiu de caz: sentimentul tweet-urilor financiare
- Transformerele în practică
- Concluzii

Codificarea sinusoidală a poziției



Sursa: Vaswani et al. (2017), formula sinusoidală. Date: ilustrarea mecanismului, $d = 64$, 64 de poziții.

- **Formula**: $p_{t,2i} = \sin(t/10000^{2i/d})$, $p_{t,2i+1} = \cos(t/10000^{2i/d})$, adunată la reprezentarea tokenului
  - Dimensiunile mici se schimbă repede, cele mari încet
- **De ce**: auto-atenția este echivariantă la permutări
  - Fără poziții, "dobînzile scad, acțiunile cresc" este totuna cu "acțiunile scad, dobînzile cresc"

DATE SIMULATE (doar pentru ilustrare) (Vaswani et al., 2017) NN_L7_positional_encoding

Un bloc transformer: atenție, rețea directă, conexiuni reziduale, normalizare

⊡ **Pe scurt:** Atenția amestecă informația între cuvinte; o rețea mică cu două straturi transformă apoi fiecare cuvînt separat; conexiunile reziduale și normalizarea pe strat stabilizează antrenarea.

$$Z = X + \mathrm{MHA}\left(\mathrm{LN}(X)\right), \qquad X' = Z + \mathrm{FFN}\left(\mathrm{LN}(Z)\right)$$

$$\mathrm{FFN}(\mathbf{z}) = W_2\ \mathrm{GELU}(W_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2, \quad \mathrm{LN}(\mathbf{z}) = \gamma \odot \frac{\mathbf{z} - \mu_{\mathbf{z}}}{\sigma_{\mathbf{z}}} + \beta$$

⊡ **unde**

- Varianta cu normalizare înainte
- Rețeaua directă aplicată fiecărei poziții (FFN), $W_1 \in \mathbb{R}^{4d \times d}$
- Unitatea liniară cu eroare gaussiană (GELU) sau unitatea liniară rectificată (ReLU)
- Normalizarea pe strat (LN), pe cele d variabile ale unui token
- Parametri pe bloc $\approx 12d^2$.

⊡ **Conexiunile reziduale**

- Gradientul ajunge direct la primele straturi (cursul 4)

⊡ **Blocul nostru**

- $d = 64$, lățimea FFN 128, dropout 0,1; 2 blocuri: 67 139 de parametri în afara reprezentărilor

⊡ **În practică:** cei mai mulți parametri ai unui transformer se află în straturile directe, nu în atenție.

(Vaswani et al., 2017; Ba, Kiros & Hinton, 2016)

Exercițiu practic: notebook, secțiunea 3

Codificatoare și decodoare: două obiective de preantrenare

- **Pe scurt:** Același bloc devine cititor sau scriitor în funcție de mască și de sarcina de preantrenare.

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in \mathcal{M}} \log p_\theta \left(x_i \mid x_{\setminus \mathcal{M}} \right)$$

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{n} \log p_\theta \left(x_t \mid x_{<t} \right)$$

- **unde**
  - $\mathcal{M} = 15\,\%$ dintre poziții, mascate aleator: modelarea limbajului cu cuvinte mascate (MLM), bidirecțională
  - Dreapta: prezicerea tokenului următor, cu mască cauzală, $M_{ij} = -\infty$ pentru $j > i$.

- **Doar codificator**
  - Bidirectional Encoder Representations from Transformers (BERT): clasificare, extragere de informații
- **Doar decodor**
  - Generative Pre-trained Transformer (GPT): generare de text, modelele mari de limbaj (LLM) de azi

- **În practică:** pentru etichetarea textelor folosiți un codificator
  - Pentru redactarea rapoartelor, un decodor (cursul 12).

(Devlin et al., 2019; Radford et al., 2018)

Tokenizarea textului în subcuvinte

- **Pe scurt:** Un vocabular fix de bucăți de cuvinte acoperă orice cuvînt, inclusiv simbolurile bursiere și numele noi de firme, împărțind cuvintele rare în bucăți frecvente.

$$x_{1:n} = \Big[\texttt{[CLS]}, w_1, \ldots, w_{n-2}, \texttt{[SEP]}\Big], \qquad \mathbf{x}_t = E[x_t] + \mathbf{p}_t + \mathbf{s}_t$$

- **unde**
  - $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ = tabelul reprezentărilor
  - $\mathbf{p}_t$ = poziția, $\mathbf{s}_t$ = segmentul
  - CLS rezumă textul, [SEP] îl închide
  - Vocabularul WordPiece al FinBERT: $|\mathcal{V}| = 30\,522$.

- **Tweet-urile noastre**
  - Mediana 25 de tokeni, percentila 99 este 57, maximul 82
  - Tăiem la 64 de tokeni: 0,19 % dintre tweet-uri pierd cîțiva tokeni
- **Reprezentările domină modelele mici**
  - Transformerul mic: 2 020 547 de parametri, dintre care 67 139 în afara tabelului

- **În practică:** verificați cum împarte tokenizatorul simbolurile bursiere și numele produselor înainte de a avea încredere în model.

(Wu et al., 2016; Devlin et al., 2019)

Exercițiu practic: notebook, secțiunea 1

Atenția, recurența și convoluția comparate

Tipul stratului	Cost pe strat	Pași secvențiali	Drumul cel mai lung
Auto-atenție	$O(n^2 d)$	$O(1)$	$O(1)$
Recurent (RNN, LSTM)	$O(nd^2)$	$O(n)$	$O(n)$
Convoluție, nucleu k	$O(knd^2)$	$O(1)$	$O(\log_k n)$

- **Rețeaua neuronală recurentă (RNN)**: ieftină pe pas, lentă pe propoziție, drumuri lungi
  - **rețeaua neuronală convoluțională (CNN)**: paralelă, dar are nevoie de adîncime ca să lege cuvinte îndepărtate
- **Auto-atenția**: drum constant și paralelism complet, plătite cu memorie n^2
  - Pentru tweet-uri ($n \approx 25$) termenul pătratic este neglijabil

(Vaswani et al., 2017, Table 1)

Cuprins

- Motivația mecanismului de atenție
- Mecanismul atenției
- Arhitectura transformer
- Studiu de caz: sentimentul tweet-urilor financiare
- Transformerele în practică
- Concluzii

Șapte modele evaluate pe același set de test

Model	Ce se antrenează
TF-IDF + logistic	regresie logistică pe secvențe de 1–2 cuvinte (17 119 de variabile), C ales pe validare
Medie de embedding-uri	media reprezentărilor tokenilor, fără atenție
Transformer mic	2 blocuri, 4 capete, $d = 64$, de la zero; și fără poziții
FinBERT fără antrenare	BERT antrenat suplimentar pe texte financiare (FinBERT, Araci, 2019), folosit ca atare
FinBERT înghețat + logistic	regresie logistică pe vectorul mediu din ultimul strat (768)
FinBERT ajustat	toate cele 109,5 milioane de ponderi, 3 epoci

- **Împărțirea**: tweet-urile de antrenare publicate = 8 588 de antrenare + 955 de validare (10 %, stratificat)
  - Setul de validare publicat = testul nostru (2 388), folosit o singură dată
- **De ce nu Financial PhraseBank**: FinBERT a fost ajustat pe aceste date, deci un test acolo ar fi contaminat
  - Tweet-urile sînt noi pentru toate modelele

(Araci, 2019; Malo et al., 2014)

Exercițiu practic: notebook, secțiunea 5

Transformerul nostru mic, antrenat de la zero

- **Pe scurt:** Două blocuri peste reprezentările tokenilor și ale pozițiilor, media pe tokenii reali, apoi un strat liniar spre cele trei clase.

$$\mathbf{h} = \frac{1}{n}\sum_{t=1}^{n} \mathbf{x}_t^{(2)}, \qquad \mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} \log \operatorname{softmax}(W\mathbf{h}_i + \mathbf{b})_{y_i}$$

- **unde**
  - $\mathbf{x}_t^{(2)}$ = ieșirea celui de-al doilea bloc pentru tokenul t
  - w_c = ponderea clasei, invers proporțională cu frecvența ei
  - Adam cu penalizare decuplată a ponderilor (AdamW), rata de învățare $5 \cdot 10^{-4}$, lot 64, oprire timpurie după macro-F1 pe validare.

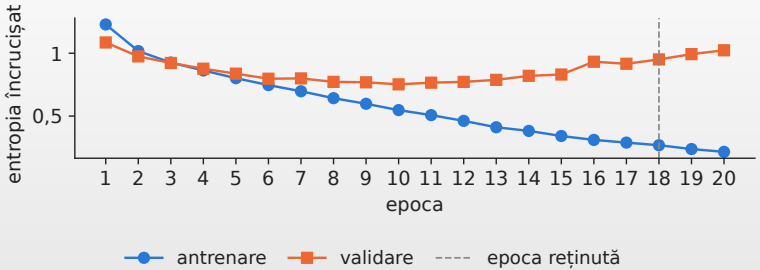
- **Dimensiunea**
  - 2 020 547 de parametri, mai ales tabelul reprezentărilor
- **Costul**
  - 44 s pe sămînță; în mediană 20 epoci pînă la oprire
- **Macro-F1**
  - Media celor trei scoruri F1 pe clase: clasa negativă contează cît cea neutră

- **În practică:** macro-F1 este măsura potrivită cînd clasa rară (știrile negative) este cea care costă bani.

(Loshchilov & Hutter, 2019)

Exercițiu practic: notebook, secțiunea 5

Transformerul mic supraînvață: pierderea pe validare crește după cîteva epoci

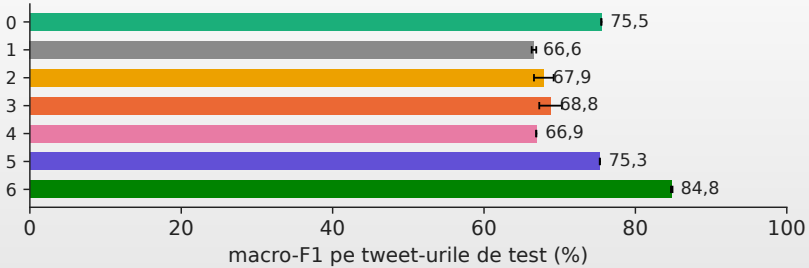


Sursa: Seturile Twitter financial news (Hugging Face, zeroshot). Date: estimări proprii; transformer mic, sămînța 1.

- **Sămînța 1**: pierderea pe validare este minimă la epoca 10 (0,753) și urcă la 1,024 pînă la epoca 20
  - Macro-F1 folosit pentru oprire atinge totuși maximul la epoca 18
- **Are nevoie de multe date**: 8 588 de tweet-uri sînt puține pentru 2 020 547 de parametri învățați de la zero
  - Modelul devine prea sigur pe el înainte să se oprească din progres

NN_L7_tiny_learning_curve

Compararea modelelor: macro-F1 pe setul de test



Sursa: Seturile Twitter financial news (Hugging Face, zeroshot). Date: estimări proprii; 2.388 tweet-uri de test (setul de validare publicat).

- **Macro-F1 pe 2 388 de tweet-uri de test**: FinBERT ajustat 84,8 %, TF-IDF 75,5 %, transformerul mic 68,8 % (abatere standard 1,5, 3 semințe)
- **FinBERT fără antrenare pe tweet-uri**: 66,9 %
  - FinBERT înghețat + logistic: 75,3 %

NN_L7_model_comparison

Clasa negativă este cea în care modelele diferă cel mai mult

Model	Macro-F1	Acuratețe	Recall negativ	Precizie negativ	Rulări
TF-IDF + logistic	75,5	81,5	70,3	63,5	1
Medie embedding-uri	66,6	74,3	62,0	46,6	3
Transformer fără poziții	67,9	75,4	61,8	48,4	3
Transformer mic	68,8	76,7	59,7	51,1	3
FinBERT fără antrenare	66,9	72,6	76,1	49,4	1
FinBERT înghețat + logistic	75,3	80,1	78,7	58,6	1
FinBERT ajustat	84,8	88,2	82,9	77,9	3

- **Toate valorile în %**, pe tweet-urile de test
  - Verde = cel mai bun, roșu = cel mai slab din coloană
  - Medii pe rulări
- **Știrile negative**: FinBERT ajustat găsește 82,9 % dintre ele, cu o precizie de 77,9 %
  - TF-IDF 70,3 % și 63,5 %

- **În practică:** raportați recall-ul și precizia pe clase: o medie poate ascunde un model care ratează știrile importante.

(calcule proprii)

Exercițiu practic: notebook, secțiunea 5

Testul bootstrap pereche al diferențelor pe tweet-urile de test

- **Pe scurt:** Reeșantionînd tweet-urile de test, cu aceeași extragere pentru ambele modele, vedem cît s-ar putea mișca întîmplător o diferență de macro-F1.

$$\Delta^{*b} = F_1^A(\mathcal{T}^{*b}) - F_1^B(\mathcal{T}^{*b}), \quad b = 1, \ldots, 1000$$
$$\text{CI}_{95\%} = [q_{0.025}, q_{0.975}]$$

- **unde**
  - $\mathcal{T}^{*b}$ = tweet-uri de test extrase cu întoarcere
  - F_1 = macro-F1 mediat pe semințele modelului
  - Diferențe în puncte procentuale.

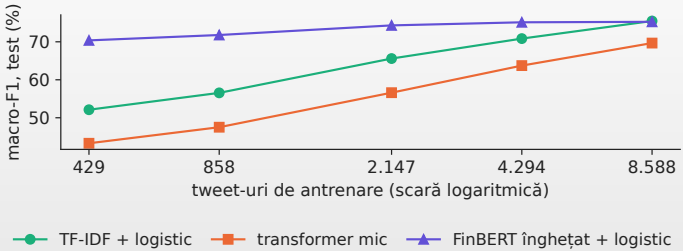
- **FinBERT ajustat − TF-IDF**
  - +9,3 [+7,3; +11,3]
- **Transformer mic − TF-IDF**
  - −6,6 [−8,3; −4,9]
- **Cu − fără poziții**
  - +0,9 [+0,1; +1,7]
- **FinBERT înghețat − TF-IDF**
  - −0,2 [−2,5; +2,0]

- **În practică:** un interval care nu conține zero este dovada minimă înainte de a înlocui un model aflat în producție.

(Efron & Tibshirani, 1993)

Exercițiu practic: notebook, secțiunea 5

Cu puține date etichetate, reprezentările preantrenate contează cel mai mult



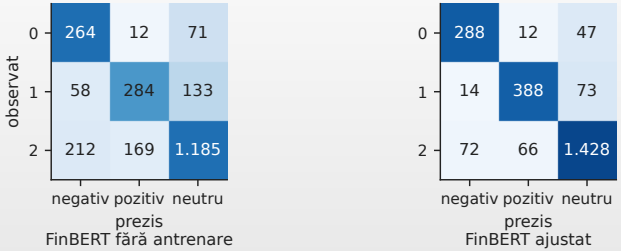
Sursa: Seturile Twitter financial news (Hugging Face, zeroshot). Date: estimări proprii; 2.388 tweet-uri de test (setul de validare publicat).

- **Cu 429 de tweet-uri de antrenare**: FinBERT înghețat + logistic 70,4 %, TF-IDF 52,1 %, transformerul mic 43,3 % macro-F1
- **Preantrenarea = date pe care nu le-ați etichetat**: miliarde de cuvinte au format deja vectorii FinBERT
  - Un model de la zero trebuie să învețe deodată limba și sarcina

(calcule proprii, sămînța 1)

NN_L7_data_scaling

Erorile modelului fără antrenare corectate prin ajustare fină



Sursa: Seturile Twitter financial news (Hugging Face, zeroshot). Date: estimări proprii; 2.388 tweet-uri de test (setul de validare publicat).

- **Fără antrenare**: FinBERT a fost antrenat pe propoziții din știri, nu pe tweet-uri
  - Găsește 264 dintre cele 347 de tweet-uri negative
  - După ajustare: 288
- **Erorile rămase**: 47 tweet-uri negative și 73 pozitive citite ca neutre
  - Doar 12 negative citite ca pozitive

(calcule proprii, modelul ajustat cu sămînța 1)

NN_L7_confusion

Ajustarea fină: ponderi inițiale preantrenate și rate de învățare mici

- **Pe scurt:** Toate ponderile se mișcă, dar puțin: cunoștințele de limbă rămîn, stratul de decizie se adaptează la tweet-uri.

$$\theta_0 = \theta_{\text{FinBERT}}, \qquad \theta_{s+1} = \theta_s - \eta_s \widehat{\nabla \mathcal{L}}(\theta_s), \quad \eta_s = \eta_0 \left(1 - \frac{s}{S}\right)$$

- **unde**
  - $\eta_0 = 2 \cdot 10^{-5}$, scădere liniară pe $S = 3$ epoci cu loturi de 32
  - AdamW cu penalizarea ponderilor 0,01
  - Se păstrează epoca cu cel mai bun macro-F1 pe validare.

- **Cîștigul**
  - +17,9 puncte față de varianta fără antrenare, +9,3 față de TF-IDF
- **Costul**
  - 109,5 milioane de ponderi, 5,2 minute pe sămînță pe procesorul grafic al unui laptop
- **Riscul**
  - O rată de învățare mare șterge preantrenarea (uitare catastrofală)

- **În practică:** ajustarea fină a unui codificator preantrenat este azi alegerea implicită pentru orice sarcină pe text etichetat, cu cîteva mii de exemple.

(Devlin et al., 2019; Araci, 2019)

Exercițiu practic: notebook, secțiunea 5

Rolul ordinii cuvintelor în clasificarea tweet-urilor

- **Experimentul de eliminare**
  - Același transformer mic, cu și fără codificarea poziției, cîte 3 semințe
  - Macro-F1 68,8 % cu poziții, 67,9 % fără: $+0{,}9$ puncte
- **Bootstrap**
  - Intervalul de 95 % al cîștigului: $[+0{,}1;\ +1{,}7]$ puncte
- **Interpretare**
  - Tweet-urile scurte și formulaice sînt mai ales colecții de cuvinte-semnal (“cut”, “beats”, “downgrade”)
  - Media de embedding-uri, fără atenție, este aproape: 66,6 %
- **Cînd contează ordinea**
  - Negația și contrastul (“not”, “but”), documentele lungi, cifrele legate de entitatea corectă

- **În practică:** testați mai întîi versiunea simplă a modelului: pe texte scurte o mare parte din puterea transformerului rămîne nefolosită.

(calcule proprii)

Exercițiu practic: notebook, secțiunea 5

Cuprins

- Motivația mecanismului de atenție
- Mecanismul atenției
- Arhitectura transformer
- Studiu de caz: sentimentul tweet-urilor financiare
- Transformerele în practică
- Concluzii

Scorurile de sentiment ca semnal de afaceri

- **Fluxul de lucru**
  - Fluxul de știri și postări → tokenizator → codificator ajustat → probabilitățile claselor
  - Agregare pe firmă și zi: ponderea mesajelor negative minus pozitive, ponderată cu audiența
- **Utilizări**
  - Monitorizarea riscului unui portofoliu de credite, avertizare timpurie pentru emitenți, o intrare pentru ajustarea valorii la risc (VaR)
- **Capcane**
  - Informația din viitor: folosiți doar texte cu marcaj de timp anterior deciziei
  - Schimbarea domeniului: un model pentru propoziții din știri citește greșit tweet-urile (diferența față de varianta fără antrenare)
  - Zgomotul din etichete și sarcasmul; derivă cînd se schimbă vocabularul pieței

- **În practică:** un scor de sentiment este o intrare într-un model, nu o decizie de tranzacționare: testați-l retrospectiv înainte de folosire.

(Tetlock, 2007)

Documentele lungi: discursurile băncilor centrale și rapoartele anuale

- **Discursurile noastre**
  - 400 de discursuri ale BCE, 8 ianuarie 2022 – 30 octombrie 2026
  - Mediana 3 338 de tokeni, cel mai lung 22 646
- **Costul pătratic**
  - Atenția pe un discurs median costă de circa 17 828 de ori mai mult decît pe un tweet median
- **Variante**
  - Împărțire în ferestre de 512 și agregarea scorurilor pe bucăți
  - Modele ierarhice: vectori de propoziții, apoi atenție pe propoziții
  - Modele cu context lung: atenție rară sau eficientă ca memorie

- **În practică:** pentru o echipă de politică monetară, împărțirea în bucăți plus un codificator bine testat bate de obicei un model cu context lung netestat.

(Vaswani et al., 2017)

Exercițiu practic: notebook, secțiunea 1

Transformerele pentru alte date decît textul

- **Serii de timp**
  - Fragmente ale seriei ca tokeni; modele de prognoză preantrenate (cursul 13)
- **Tabele și tranzacții**
  - Fiecare coloană sau fiecare plată anterioară ca token; atenție între ele
- **Grafuri**
  - Atenția restrînsă la muchii = rețeaua cu atenție pe graf (cursul 9)
- **Imagini și documente**
  - Fragmente de imagine ca tokeni; modelele viziune-limbaj citesc grafice și rapoarte (cursul 14)

- **În practică:** oricare ar fi datele, întrebarea este aceeași: care sînt tokenii și ce perechi au voie să-și acorde atenție?

(Bishop & Bishop, 2024, ch. 12)

Cuprins

- Motivația mecanismului de atenție
- Mecanismul atenției
- Arhitectura transformer
- Studiu de caz: sentimentul tweet-urilor financiare
- Transformerele în practică
- Concluzii

Idei principale

- **Atenția** = softmax al scorurilor întrebare-cheie scalate, înmulțit cu valorile
  - Un singur pas între oricare două cuvinte, cost $O(n^2)$
  - $\sqrt{d_k}$ păstrează softmax-ul antrenabil
- **Blocul transformer** = atenție cu mai multe capete + rețea directă, fiecare cu conexiune reziduală și normalizare
  - Pozițiile se adaugă, pentru că atenția ignoră ordinea
- **Pe tweet-urile financiare** preantrenarea decide: FinBERT ajustat 84,8 % macro-F1, TF-IDF 75,5 %, un transformer de la zero 68,8 %
- **Hărțile de atenție** sînt indicii, nu explicații
  - Verificați erorile pe clase și intervalele bootstrap înainte de a avea încredere într-un model

Cercetările grupului nostru: transformerele ca instrumente de prognoză a riscului

- **Modele mari de limbaj (LLM) pentru prognoza riscului extrem** (Pele, Bolovăneanu, Lin, Ren, Ginavar, Spilak, Andrei, Toma, Lessmann & Härdle, 2026)
  - Lucrarea folosește transformere decodor, cărora li se dau istoricele randamentelor, pentru prognoza valorii la risc și a pierderii așteptate în coadă (ES) — LLM-VaR și LLM-ES — și le evaluează cu testele retrospective standard
- **Modele fundaționale pentru riscul extrem** (Jheng, Tak, Găman, Mazurencu-Marinescu-Pele & Pele, 2026)
  - Lucrarea folosește modele transformer de prognoză preantrenate, fără reantrenare și cu calibrare conformală, pentru prognoza VaR și ES pe piețele din Europa Centrală și de Est
- **Legătura cu acest curs**: aceeași idee ca la FinBERT, un model preantrenat pe date multe și folosit cu puțină sau fără antrenare specifică, aplicată măsurilor de risc în loc de cuvinte

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)

Explorați vizual

- **Transformer Explainer**: un model GPT-2 care rulează în browser: scrieți o propoziție, deschideți un cap de atenție și urmăriți scorurile $\mathbf{q}^\top \mathbf{k}$, softmax-ul și suma ponderată a valorilor
  - Mutați temperatura ca distribuția următorului token să devină mai ascuțită sau mai plată.
- **LLM Visualization (B. Bycroft)**: o parcurgere 3D a unui GPT mic, bloc cu bloc: fiecare produs de matrice din normalizare, atenție și stratul feed-forward.
- **The Illustrated Transformer (J. Alammar)**: queries, keys și values desenate pas cu pas pentru o propoziție.
- **Variante ale atenției (S. Raschka)**: de la atenția multi-head la grouped-query și latent attention, ferestre glisante și atenție rară: cum reduc modelele recente memoria pentru keys și values.

Lecturi și resurse suplimentare

- **Vaswani et al. (2017)**, "Attention is all you need": transformerul original
- **Bishop & Bishop (2024)**, capitolul 12: transformerele, de la atenție la modelele mari de limbaj
- **The Illustrated Transformer** (J. Alammar): jalammar.github.io.
- **Cursul Hugging Face**, ajustarea codificatoarelor: huggingface.co/learn.
- **Acest curs**: cod, date, notebook-uri — github.com/danpele/NEURAL_BIZ.
- **Exerciții practic**: notebook-ul reproduce toate graficele și cifrele acestui curs

Test 1: scalarea

- De ce se împart scorurile de atenție la $\sqrt{d_k}$?

Răspuns

- Cu elemente de varianță unu, produsul scalar $\mathbf{q}^\top \mathbf{k}$ are varianța d_k
  - Împărțirea la $\sqrt{d_k}$ o readuce la 1, deci softmax-ul nu se saturează și gradientul nu dispare

Test 2: pozițiile

- Ce se întîmplă cu un transformer fără codificarea poziției?

Răspuns

- Auto-atenția este echivariantă la permutări, deci modelul vede o colecție de tokeni
  - Pe tweet-urile noastre scurte costul a fost doar 0,9 puncte de macro-F1, dar ordinea contează pentru negație și texte lungi

Test 3: preantrenarea

- De ce pierde în fața TF-IDF un transformer antrenat de la zero pe 8 588 de tweet-uri?

Răspuns

- Trebuie să învețe limba și sarcina din puține exemple, cu 2 020 547 de parametri
  - FinBERT preantrenat aduce aceste cunoștințe și ajunge la 84,8 % după ajustare

Test 4: codificator sau decodor

- O bancă vrea să eticheteze după subiect 10.000 de reclamații pe zi.
  - Transformer codificator sau decodor?

Răspuns

- Un codificator (de tip BERT), ajustat pe reclamații etichetate: citește tot textul în ambele direcții și produce probabilitățile claselor, ieftin și consecvent

Test 5: hărțile de atenție

- Poate o hartă de atenție să justifice o decizie în fața unui supraveghetor?

Răspuns

- Nu: hărți diferite pot da aceeași predicție, iar multă atenție merge spre [SEP] (pînă la 59,3 % aici)
  - Se folosesc metode de atribuire și teste

Bibliografie (1/2)

- Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. arXiv:1908.10063
- Ba, J. L., Kiros, J. R. & Hinton, G. E. (2016). Layer normalization. arXiv:1607.06450
- Bahdanau, D., Cho, K. & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*. arXiv:1409.0473
- Bishop, C. M. & Bishop, H. (2024). *Deep Learning: Foundations and Concepts*, ch. 12. Springer. doi:10.1007/978-3-031-45468-4
- Clark, K., Khandelwal, U., Levy, O. & Manning, C. D. (2019). What does BERT look at? An analysis of BERT's attention. *BlackboxNLP Workshop*, 276–286. doi:10.18653/v1/W19-4828
- Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL*, 4171–4186. doi:10.18653/v1/N19-1423
- Efron, B. & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall. doi:10.1201/9780429246593
- Jain, S. & Wallace, B. C. (2019). Attention is not explanation. *Proceedings of NAACL*, 3543–3556. doi:10.18653/v1/N19-1357

Bibliografie (2/2)

- 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
- Loshchilov, I. & Hutter, F. (2019). Decoupled weight decay regularization. *International Conference on Learning Representations*. arXiv:1711.05101
- Malo, P., Sinha, A., Korhonen, P., Wallenius, J. & Takala, P. (2014). Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4), 782–796. doi:10.1002/asi.23062
- 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
- Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI technical report. openai.com/research/language-unsupervised
- Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, 62(3), 1139–1168. doi:10.1111/j.1540-6261.2007.01232.x
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. arXiv:1706.03762
- Wu, Y., Schuster, M., Chen, Z. et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144

Anexa A: gradientul softmax-ului și saturația

$$a_j = \frac{e^{s_j}}{\sum_l e^{s_l}}, \qquad \frac{\partial a_j}{\partial s_i} = a_j(\delta_{ij} - a_i)$$

$$J = \text{diag}(\mathbf{a}) - \mathbf{a}\mathbf{a}^\top, \qquad \|J\| \to 0 \text{ cînd } \mathbf{a} \to \mathbf{e}_k$$

- Dacă un scor domină, $\mathbf{a}$ este aproape one-hot și toate elementele lui J sînt aproape zero: întrebările și cheile nu mai învață
- Scorurile cu abaterea standard σ fac dominarea probabilă cînd σ este mare
  - Scalarea cu $\sqrt{d_k}$ fixează $\sigma = 1$ la inițializare

(Vaswani et al., 2017)

Anexa B: parametrii unui bloc transformer

$$\underbrace{4d^2 + 4d}_{\text{atenție}} + \underbrace{8d^2 + 5d}_{\text{FFN}} + \underbrace{4d}_{\text{LN}} \approx 12d^2$$

- Atenția: $W_Q, W_K, W_V, W_O \in \mathbb{R}^{d \times d}$, cu termeni liberi
  - FFN: $W_1 \in \mathbb{R}^{4d \times d}$, $W_2 \in \mathbb{R}^{d \times 4d}$
  - Două normalizări cu γ, β
- BERT-base, $d = 768$: circa 7 milioane pe bloc, 85 de milioane în 12 blocuri, plus 23 de milioane în tabelul reprezentărilor (după datele publicate)

(Devlin et al., 2019)

Anexa C: setările experimentelor

Model	Setări
TF-IDF	secvențe de 1–2 cuvinte, minimum 2 documente, frecvență sublineară; regresie logistică cu ponderi de clasă echilibrate, $C = 3$ ales pe validare
Transformer mic	tokenii WordPiece ai FinBERT, 64 de tokeni; $d = 64$, 4 capete, 2 blocuri cu normalizare înainte, FFN 128, dropout 0,1; AdamW $5 \cdot 10^{-4}$, lot 64, cel mult 20 de epoci, răbdare 4
FinBERT ajustat	versiunea ProsusAI, cu stratul de 3 clase reasociat claselor negativ/pozitiv/neutru; AdamW $2 \cdot 10^{-5}$, scădere liniară, lot 32, 3 epoci; semințele 1–3
FinBERT înghețat	media ultimului strat ascuns pe tokenii reali; regresie logistică, $C = 0{,}03$
Echipament	laptop cu procesor Apple, procesorul grafic, Metal Performance Shaders (MPS); rularea completă 8,1 minute