

TODO Zahvala

Sadržaj

1. Uvod	3
2. Metodologija	4
2.1. Povrati imovine	4
2.2. Statistički modeli povrata	7
2.3. Faktorski modeli	11
2.3.1. Linearni faktorski model	12
2.3.2. Jednofaktorski model povrata	14
2.4. Duboko učenje	14
2.4.1. Povratne neuronske mreže	16
2.4.2. LSTM mreže	18
2.4.3. Treniranje modela dubokog učenja	21
3. Implementacija	23
3.1. Podatci	23
3.2. Obrada podataka	25
3.3. Sintetički podatci	26
3.4. Model dubokog učenja	27
4. Rezultati	32
5. Zaključak	33
Literatura	34
Sažetak	36

Abstract	37
---------------------------	-----------

1. Uvod

TODO Uvod.

2. Metodologija

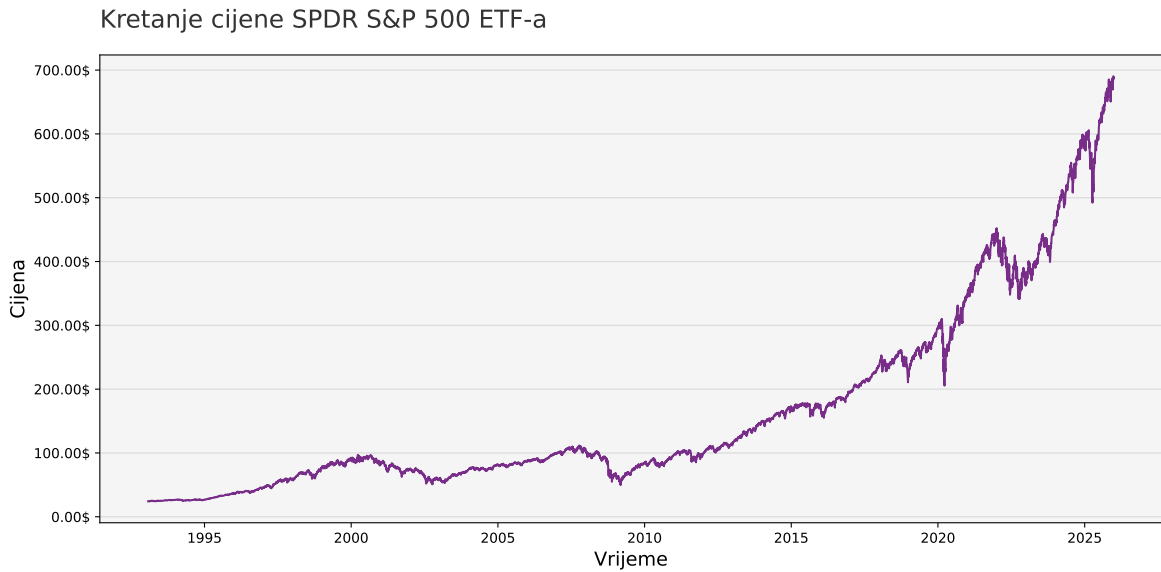
Ovo poglavlje pokriva teoretsku osnovu potrebnu za razumijevanje implementiranog modela dubokog učenja te strukturu podataka na kojima se primjenjuje. Prvo ćemo definirati osnovne varijable koje ćemo se oslanjati u ovom radu. Zatim ćemo proučiti njihova statistička svojstva. Nakon toga razmotrit ćemo faktorske modele koji omogućuju modeliranje naših varijabli. Na kraju ćemo pokriti duboko učenje i specifičnu arhitekturu koja će se koristiti u ovom radu.

2.1. Povrati imovine

Vrijednosni papiri predstavljaju financijske instrumente koji potvrđuju određena imovinska ili druga prava njihova vlasnika, poput prava na udio u vlasništvu poduzeća (dionice) ili prava na povrat uložених sredstava uz kamatu (obveznice). Njima se trguje na organiziranim tržištima kapitala, poput burzi, gdje se kupnja i prodaja odvijaju putem ovlaštenih posrednika, a cijene se formiraju na temelju ponude i potražnje. Tržišna cijena vrijednosnog papira u svakom trenutku odražava ravnotežu između kupaca i prodavatelja, pri čemu se svaka realizirana transakcija bilježi kao nova referentna cijena. Tržišni indeksi predstavljaju promjene vrijednosti grupe dionica ili drugih financijskih instrumenata. Služe kao mjerilo performansi tržišta ili određenog segmenta tržišta [1].

Kako bi se omogućila analiza kretanja cijena kroz vrijeme, kontinuirani tok transakcijskih podataka uzorkuje se u diskretnim vremenskim intervalima (npr. minute, sati, dani), čime se dobivaju vremenski nizovi cijena koji služe kao temelj za statističku analizu i modeliranje. Slika 2.1. prikazuje kretanje dnevnih cijena za SPDR S&P 500 ETF (engl. *exchange traded fund*) od njegovog začeća 1993. do kraja 2025. godine.

U ovom radu koristit ćemo povrate umjesto samih cijena imovina. Postoje dva glavna



Slika 2.1. Kretanje cijene SPDR S&P 500 ETF-a

razloga za to. Prvi je to što povrati sažimaju kretanja cijena te ih stavljaju na jednaku i usporedivu skalu. Drugi se odnosi na povoljnija statistička svojstva povrata u odnosu na cijene. Postoji više načina za definirati povrate [2].

Jednostavni (artimetički) povrati

Nek je P_t cijena imovine u trenutku t . Ako imovinu posjedujemo od trenutka $t - 1$ do trenutka t , ostvareni artimetički povrat dobivamo izrazom:

$$1 + R_t = \frac{P_t}{P_{t-1}} \quad (2.1)$$

$$R_t = \frac{P_t}{P_{t-1}} - 1 = \frac{P_t - P_{t-1}}{P_{t-1}} \quad (2.2)$$

Ukoliko imovinu držimo kroz T perioda ukupni artimetički povrat dobivamo ukupnim množenjem artimetičkih povrata u vremenu [3]:

$$\begin{aligned} 1 + R_{total} &= \frac{P_T}{P_1} = \frac{P_T}{P_{T-1}} \times \frac{P_{T-1}}{P_{T-2}} \times \dots \times \frac{P_2}{P_1} \\ &= (1 + R_T)(1 + R_{T-1}) \dots (1 + R_1) \\ &= \prod_{t=1}^T (1 + R_t). \end{aligned} \quad (2.3)$$

Kontinuirani (logaritamski) povrati

Prirodni logaritam artimetričkog povrata (2.1) je logaritamski povrat:

$$r_t = \ln(1 + R_t) = \ln \frac{P_t}{P_{t-1}} = p_t - p_{t-1}, \quad (2.4)$$

gdje $p_t = \ln(P_t)$.

Razmotrimo zatim ukamaćivanje logaritamskih povrata u vremenu:

$$\begin{aligned} r_{total} &= \ln(1 + R_{total}) = \ln[(1 + R_T)(1 + R_{T-1}) \dots (1 + R_1)] \\ &= \ln(1 + R_T) + \ln(1 + R_{T-1}) + \dots + \ln(1 + R_1) \\ &= r_T + r_{T-1} + \dots + r_1 \\ &= \sum_{t=1}^T r_t \end{aligned} \quad (2.5)$$

Ukamaćivanje logaritamski povrata može se ostvariti zbrajanjem pojedinačnih logaritamskih povrata. To svojstvo zovemo *aditivnost u vremenu*. Logaritamski povrati također posjeduju poželjna statistička svojstva [3]. Za male magnitude vrijedi $r_t \approx R_t$. Ova aproksimacija je korisna kada se razmatraju kratki vremenski intervali [1]. Slika 2.2. prikazuje dnevne artimetričke i logaritamske povrate SPDR S&P 500 ETF. Sa slike vidimo da nema značajne razlike između artimetričkih i logaritamskih povrata jer su povrati izračunati na kratkom odnosno dnevnom vremenskom intervalu.

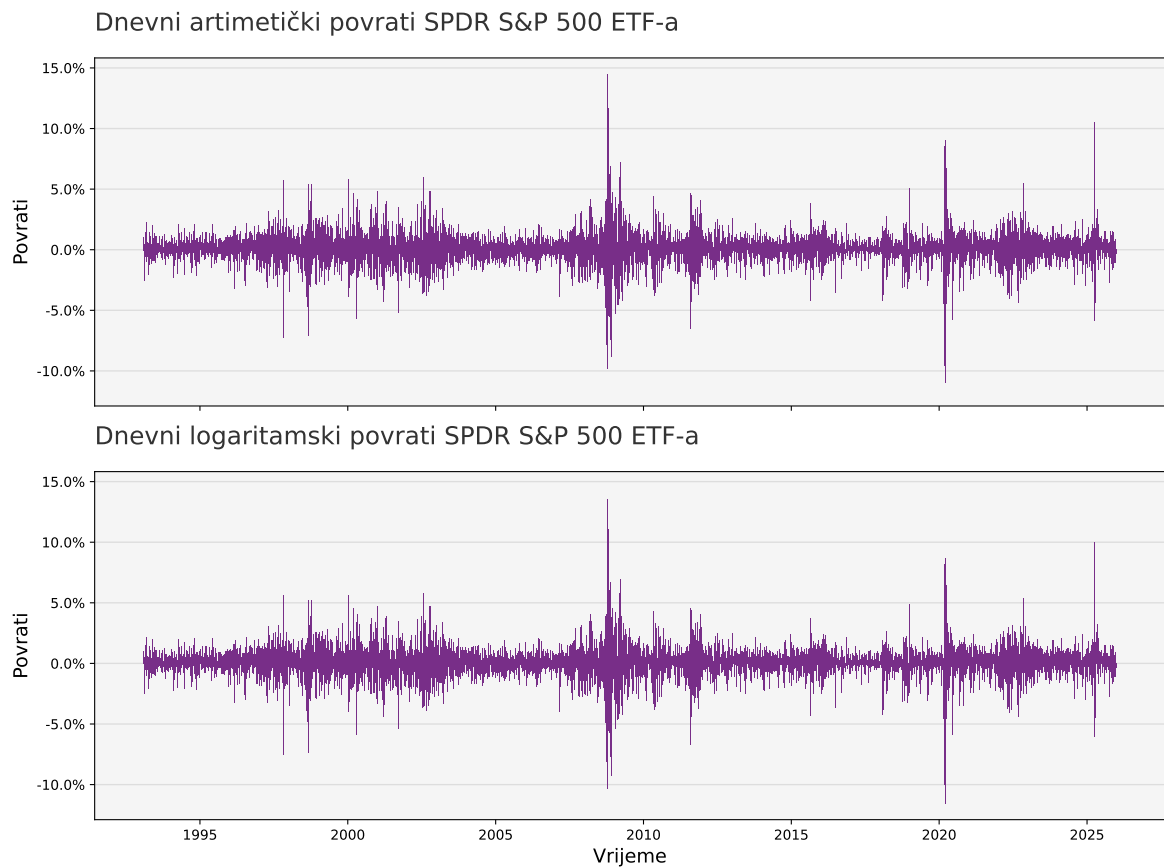
Povrat portfelja

Artimetrički povrat portfelja koji se sastoji od N vrijednosnica je otežana artimetrička sredina (engl. *weighted average*) artimetričkih povrata vrijednosnica, gdje je težina (engl. *weight*) pojedine vrijednosnice njezin udio u vrijednosti portfelja [3].

Nek je p portfelj kojem je w_i ponder vrijednosnice i . Tada je artimetrički povrat portfelja p u trenutku t dan izrazom:

$$R_{pt} = \sum_{i=1}^N w_i R_{it}, \quad (2.6)$$

gdje je R_{it} artimetrički povrat vrijednosnice i u trenutku t . Ovo svojstvo nazivamo *aditiv-*



Slika 2.2. Aritmetički i logaritamski povrati SPDR S&P 500 ETF-a

nost u prostoru vrijednosnica. Logaritamski povrati ne posjeduju to svojstvo.

Povrati iznad bezrizične kamatne stope

Povrati iznad bezrizične kamatne stope (engl. *excess return*) predstavlja razliku između ostvarenog povrata određene imovine i referentnog, bezrizičnog povrata, te mjeri dodatnu kompenzaciju koju investitor ostvaruje za preuzimanje rizika. Kao referentni povrat najčešće se koriste prinosi na državne obveznice visoke kreditne kvalitete i kratkog dospjeća, jer se smatra da nose zanemariv rizik [3].

$$R_t^{excess} = R_t - R_f \quad (2.7)$$

2.2. Statistički modeli povrata

Prije nego razmotrimo konkretne statističke modele povrata, potrebno je ukratko opisati osnovne distribucije na kojima će se temeljiti daljnja analiza te načine estimacije njihovih

parametara.

Normalna distribucija

Normalna distribucija jedan je od najčešće korištenih probabilističkih modela u statistici i financijama zbog svoje matematičke jednostavnosti i dobrih teorijskih svojstava. Ako slučajna varijabla X slijedi normalnu distribuciju sa sredinom μ i varijancom σ^2 , tada njezina funkcija gustoće ima oblik [4]:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \quad (2.8)$$

Za osmotreni uzorak $\{x_1, x_2, \dots, x_n\}$, procjena sredine dana je aritmetičkom sredinom

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i, \quad (2.9)$$

dok se varijanca procjenjuje izrazom:

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2. \quad (2.10)$$

Lognormalna distribucija

Lognormalna distribucija koristi se za modeliranje slučajnih varijabli čiji je logaritam normalno distribuiran. Ako postoji broj a takav da $Y = \ln(X - a)$ prati normalnu distribuciju, slučajna varijabla X tada slijedi lognormalnu distribuciju. Kako bi to vrijedilo, vjerojatnost da slučajna varijabla X poprimi vrijednost manju od a mora biti jednaka nuli. Ako vrijedi $Y \sim \mathcal{N}(\mu, \sigma^2)$, tada X ima gustoću:

$$f(x) = \frac{1}{(x-a)\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln(x-a)-\mu)^2}{2\sigma^2}\right], \quad x > a. \quad (2.11)$$

Sredina i varijanca lognormalne distribucije dane su izrazima

$$E[X] = e^{\mu + \frac{\sigma^2}{2}}, \quad \text{Var}[X] = (e^{\sigma^2} - 1)e^{2\mu + \sigma^2}. \quad (2.12)$$

Procjena parametara provodi se logaritamskom transformacijom uzorka, nakon čega se primjenjuju standardne procjene za normalnu distribuciju [4].

Studentova t distribucija

Studentova t distribucija predstavlja generalizaciju normalne distribucije s dodatnim parametrom broja stupnjeva slobode ν , koji kontrolira debljinu repova. Za manje vrijednosti ν distribucija ima izraženije repove, čime omogućuje robusnije modeliranje ekstremnih vrijednosti.

Gustoća Studentove t distribucije sa sredinom μ , varijancom σ^2 i ν stupnjeva slobode ima oblik:

$$f(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\nu\pi\sigma^2}} \left(1 + \frac{(x - \mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}. \quad (2.13)$$

Za procjenu parametara t distribucije koriste se metode procjene najveće izglednosti (engl. *maximum likelihood estimation*) [5].

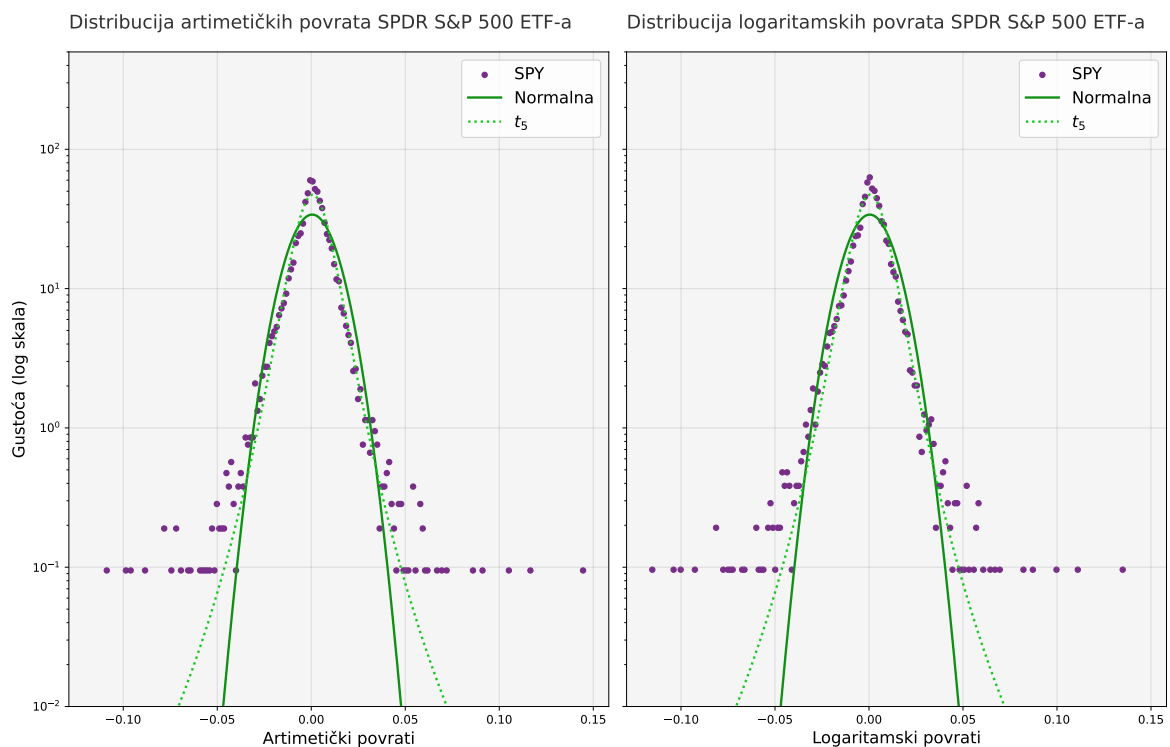
Statistička svojstva povrata

U statističkoj analizi financijskih vremenskih nizova često se polazi od pretpostavke da su aritmetički povrati nezavisne i jednako distribuirane slučajne varijable s normalnom distribucijom, konstantnom sredinom i varijancom, čime se značajno pojednostavljuje teorijska obrada i izvođenje analitičkih rezultata. Međutim, takva pretpostavka suočava se s nizom ograničenja: aritmetički povrati imaju donju granicu od -1 , dok normalna distribucija nema ograničenja po realnoj osi, višeperiodni povrati ne zadržavaju normalnu distribuciju zbog multiplikativne prirode (2.3), a empirijski podaci često pokazuju odstupanja od normalnosti [6].

Alternativno možemo pretpostaviti da su logaritamski povrati nezavisno i identično normalno distribuirani sa sredinom μ i varijancom σ^2 . S tom pretpostavkom impliciramo da su aritmetički povrati nezavisno i identično lognormalno distribuirani. Ovaj pristup ima povoljnija matematička svojstva jer su zbrojevi logaritamskih povrata kroz više razdoblja također normalno distribuirani (2.5), a pritom se prirodno zadovoljava donja granica aritmetičkog povrata (2.11). Unatoč tim prednostima, ni pretpostavka lognormalnosti u potpunosti ne opisuje realna tržišna kretanja, budući da stvarni financijski

povrati često pokazuju deblje repove distribucije i veću učestalost ekstremnih vrijednosti nego što to predviđa normalna distribucija [3].

Kako bismo bolje obuhvatili ta empirijska svojstva, možemo koristiti studentovu t -distribuciju. Slika 2.3. prikazuje distribuciju gustoće vjerojatnosti dnevnih aritmetičkih i logaritamskih povrata SPDR S&P 500 ETF-a. Također su prikazane distribucije gustoće vjerojatnosti normalne distribucije i studentove t -distribucije s pet stupnjeva slobode ¹. Parametri μ i σ^2 za normalnu i t_5 -distribuciju procijenjeni su iz aritmetičkih povrata za lijevi graf i logaritamskih povrata za desni graf. Sa slike vidimo kako t_5 -distribucija zbog svojih težih repova vijernije modelira distribuciju povrata od normalne distribucije.



Slika 2.3. Distribucija dnevnih aritmetičkih i logaritamskih povrata SPY ETF-a

¹Biramo pet stupnjeva slobode kako bi imali konačna prva četiri momenta distribucije. Za više informacija pogledati [5]

Slučajni vektori

Nek je $\mathbf{X} = (X_1, \dots, X_p)$ slučajni vektor p slučajnih varijabli. Tada su vektor sredina i kovarijacijska matrica dani izrazima:

$$E(\mathbf{X}) = \boldsymbol{\mu}_X = [E(X_1), \dots, E(X_p)]^T \quad (2.14)$$

$$\text{Cov}(\mathbf{X}) = \boldsymbol{\Sigma}_X = E[(\mathbf{X} - \boldsymbol{\mu}_X)(\mathbf{X} - \boldsymbol{\mu}_X)^T], \quad (2.15)$$

uz uvjet da dana očekivanje postoje. Neka su $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ realizacije slučajnog vektora \mathbf{X} . Tada su uzoračka sredina i kovarijacijska matrica dane izrazima [3]:

$$\hat{\boldsymbol{\mu}}_x = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t, \quad \hat{\boldsymbol{\Sigma}}_x = \frac{1}{T-1} \sum_{t=1}^T (\mathbf{x}_t - \hat{\boldsymbol{\mu}}_x)(\mathbf{x}_t - \hat{\boldsymbol{\mu}}_x)^T. \quad (2.16)$$

Ako pretpostavimo da slučajni vektor \mathbf{X} dolazi iz multivarijatne normalne distribucije s vektorom sredina $\boldsymbol{\mu}$ i kovarijacijskom matricom $\boldsymbol{\Sigma}$, tada je funkcija izglednosti uzorka (engl. *likelihood function*) dana jednadžbom [7]:

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{x}_1, \dots, \mathbf{x}_T) = (2\pi)^{-\frac{pT}{2}} |\det(\boldsymbol{\Sigma})|^{-\frac{T}{2}} \exp\left(-\frac{1}{2} \sum_{t=1}^T (\mathbf{x}_t - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_t - \boldsymbol{\mu})\right). \quad (2.17)$$

S obzirom da taj oblik funkcije izglednosti nije najprikladniji za računanje računalom iskazat ćemo i njezin logaritamski oblik (engl. *log-likelihood function*) te ćemo zamijeniti argument $\boldsymbol{\Sigma}$ s $\boldsymbol{\Sigma}^{-1}$:

$$\begin{aligned} \ln \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}; \mathbf{x}_1, \dots, \mathbf{x}_T) &= \\ &= \frac{1}{2} \left(-pT \ln(2\pi) + T \ln[\det(\boldsymbol{\Sigma}^{-1})] - \sum_{t=1}^T (\mathbf{x}_t - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_t - \boldsymbol{\mu}) \right). \end{aligned} \quad (2.18)$$

2.3. Faktorski modeli

U analizi povrata imovina često se primjenjuju multivarijatne statističke metode kako bi se proučilo ponašanje i međusobna povezanost većeg broja vrijednosnica unutar portfelja. Međutim, modeliranje velikog broja vremenskih nizova povrata dovodi do visokodimenzionalnih i kompleksnih modela koji su složeni za interpretaciju i primjenu u praksi.

Empirijska istraživanja pokazuju da se povrati različitih vrijednosnica često kreću na sličan način, što nas upućuje da postoje neki zajednički faktori koji utječu na njihovo kretanje. Primjerice, u razdobljima gospodarske krize pad aktivnosti cijelog gospodarstva obično rezultira istodobnim padom cijena većine vrijednosnica, dok rast određenog sektora, poput tehnološkog, često dovodi do rasta cijena većine dionica unutar tog sektora [8].

Te činjenice omogućuju faktorskim modelima da objasne kretanje većeg broja povrata pomoću ograničenog broja zajedničkih faktora te da pojednostavne njihovu analizu. Postoje tri vrste faktorskih modela [2]. Prvi su *markoekonomske faktorske modele* koji se fokusiraju na varijable kao što su rast BDP-a, kamatne stope, stopa inflacije i slično. U ovakvom su modelu faktori osmotrivi pa se model može estimirati linearnom regresijom. Drugi su *fundamentalni faktorske modele* koji koriste podatke o poduzećima kako bi konstruirali svoje faktore. Treći su *statistički faktorske modele* čiji su faktori neosmotrivi latentne varijable koje se estimiraju iz podataka [3]. Ovaj rad će se fokusirati prvu vrstu odnosno na *makroekonomske faktorske modele*.

2.3.1. Linearni faktorski model

Pretpostavimo da imamo p imovina kroz T vremenskih trenutaka. Neka je r_{it} povrat imovine i u trenutku t . Opća forma faktorskog modela dana je izrazom:

$$r_{it} = \alpha_i + \beta_{i1}f_{1t} + \cdots + \beta_{im}f_{mt} + \epsilon_{it}, \quad t = 1, \dots, T; \quad i = 1, \dots, p, \quad (2.19)$$

gdje je α_i konstanta, $\{f_{jt} | j = 1, \dots, m\}$ su m zajedničkih (sistemskih) faktora, β_{ij} su koeficijenti imovine i uz faktor j , a ϵ_{it} je specifičan (idiosinkratski) faktor imovine i .

Za faktor $\mathbf{f}_t = (f_{1t}, \dots, f_{mt})^T$ vrijedi:

$$E(\mathbf{f}_t) = \boldsymbol{\mu}_f, \quad (2.20)$$

$$\text{Cov}(\mathbf{f}_t) = \boldsymbol{\Sigma}_f, \quad m \times m \text{ matrica}, \quad (2.21)$$

dok je idiosinkratski faktor ϵ_{it} modeliran bijelim šumom koji nije koreliran sa sistemskim

faktorima f_{jt} i drugim idiosinkratskim faktorima. Dakle pretpostavljamo:

$$E(\epsilon_{it}) = 0 \quad \text{za sve } i, t \quad (2.22)$$

$$\text{Cov}(f_{jt}, \epsilon_{is}) = 0 \quad \text{za sve } i, j, t, s \quad (2.23)$$

$$\text{Cov}(\epsilon_{it}, \epsilon_{js}) = \begin{cases} \sigma_i^2, & \text{ako } i = j \text{ i } t = s, \\ 0, & \text{inače.} \end{cases} \quad (2.24)$$

Jednadžbu (2.19) možemo zapisati i u matričnom obliku za svih p imovina u trenutku t :

$$\mathbf{r}_t = \boldsymbol{\alpha} + \boldsymbol{\beta} \mathbf{f}_t + \boldsymbol{\epsilon}_t, \quad t = 1, \dots, T, \quad (2.25)$$

gdje je $\mathbf{r}_t = (r_{1t}, \dots, r_{pt})^T$ vektor povrata, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p)^T$ vektor konstanti, $\boldsymbol{\beta} = [\beta_{ij}]$ je $p \times m$ matrica koeficijenata, a $\boldsymbol{\epsilon}_t = (\epsilon_{1t}, \dots, \epsilon_{pt})^T$ je vektor idiosinkratskih faktora čija je kovarijacijska matrica $\text{Cov}(\boldsymbol{\epsilon}_t) = \boldsymbol{\Psi} = \text{diag}\{\sigma_1^2, \dots, \sigma_k^2\}$ $p \times p$ dijagonalna matrica. Očekivanje i kovarijacijsku matricu povrata \mathbf{r}_t možemo računati sljedećim formulama:

$$E(\mathbf{r}_t) = \boldsymbol{\alpha} + \boldsymbol{\beta} \boldsymbol{\mu}_f \quad (2.26)$$

$$\text{Cov}(\mathbf{r}_t) = \boldsymbol{\beta} \boldsymbol{\Sigma}_f \boldsymbol{\beta}^T + \boldsymbol{\Psi}. \quad (2.27)$$

Zatim jednadžbu (2.25) možemo zapisati u sljedećem obliku:

$$\mathbf{r}_t = \boldsymbol{\xi} \mathbf{g}_t + \boldsymbol{\epsilon}_t, \quad (2.28)$$

gdje $\mathbf{g}_t = (1, \mathbf{f}_t^T)^T$, a $\boldsymbol{\xi} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$ je $p \times (m + 1)$ matrica. Transponiramo li prethodnu jednadžbu i grupiramo podatke za svih T trenutaka dobivamo:

$$\mathbf{R} = \mathbf{G} \boldsymbol{\xi}^T + \mathbf{E}, \quad (2.29)$$

gdje je \mathbf{R} $T \times p$ matrica povrata čiji je t -ti redak \mathbf{r}_t^T , \mathbf{G} je $T \times (m + 1)$ matrica čiji je t -ti redak \mathbf{g}_t^T , \mathbf{E} je $T \times p$ matrica idiosinkratskih faktora čiji je t -ti redak $\boldsymbol{\epsilon}_t^T$ [3].

S obzirom da nas zanimaju *makroekonomski faktorski modeli* čiji su faktori \mathbf{f}_t osmotrivi, jednadžba (2.29) ima oblik višestruke multivariatne linearne regresije. Zbog toga

parametre modela možemo estimirati metodom najmanjih kvadrata [9]:

$$\hat{\xi}^\top = \begin{bmatrix} \hat{\alpha}^\top \\ \hat{\beta}^\top \end{bmatrix} = (\mathbf{G}^\top \mathbf{G})^{-1} (\mathbf{G}^\top \mathbf{R}), \quad (2.30)$$

odakle su α i β lako dostupni. Rezidualne, odnosno povrate idiosinkratskih faktora možemo lako dobiti koristeći formulu (2.29):

$$\hat{\mathbf{E}} = \mathbf{R} - \mathbf{G} \hat{\xi}^\top. \quad (2.31)$$

2.3.2. Jednofaktorski model povrata

Jedan od najpoznatijih *makroekonomskih faktorskih modela* koristi povrat tržišta kao faktor koji utječe na sve vrijednosnice:

$$r_{it} = \alpha_i + \beta_i r_{mt} + \epsilon_{it} \quad i = 1, \dots, p; \quad t = 1, \dots, T, \quad (2.32)$$

gdje je r_{it} povrat vrijednosnice i iznad bezrizične kamatne stope, a r_{mt} povrat tržišta iznad bezrizične kamatne stope. Kod modeliranja dionica, za povrat tržišta r_{mt} uzima se povrat nekog tržišnog indeksa (npr. S&P 500) iznad bezrizične kamatne stope. Koeficijenti modela α_i i β_i procjenjuju se metodom najmanjih kvadrata (2.30).

Ovaj rad fokusirat će se upravo na jednofaktorski model povrata. Cilj ovog rada biti će ispitati može li model dubokog učenja, iz prozora povijesnih povrata $\mathbf{R}_H = \{\mathbf{r}_t\}_{t=1}^k$, procijeniti koeficijente α_i i β_i , koji će bolje odgovarati budućem prozoru povrata $\mathbf{R}_F = \{\mathbf{r}_t\}_{t=k+1}^T$, nego procjena koeficijenata α_i i β_i koju možemo dobiti metodom najmanjih kvadrata na istom povijesnom prozoru povrata \mathbf{R}_H za $1 \leq k < T$. Način na koji ćemo mjeriti koliko dobro procjena koeficijenata α_i i β_i odgovara budućem prozoru povrata \mathbf{R}_F biti će detaljnije objašnjen u potpoglavlju 3.4.

2.4. Duboko učenje

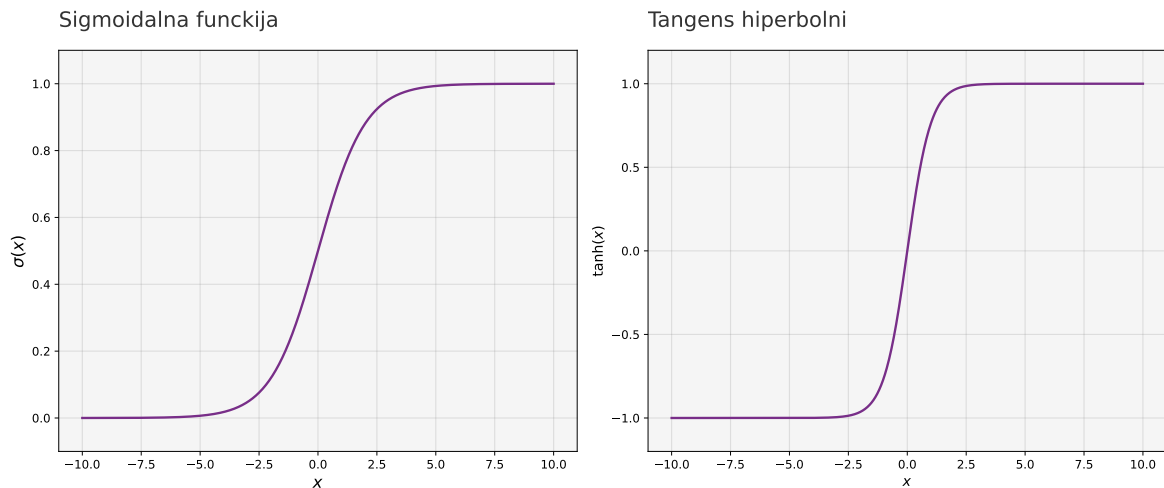
Duboko učenje predstavlja podpodručje strojnog učenja koje se ističe u rješavanju problema s visokom dimenzionalnošću podataka, kao što su računalni vid, obrada prirodnog

jezika, financijska te slične složene domene. Temeljna ideja dubokog učenja je izgradnja hijerarhijskih, složenih reprezentacija podataka, koje se dobivaju primjenom uzastopnih nelinearnih transformacija modeliranih pomoću višeslojnih neuronskih mreža. Osnovni element svake neuronske mreže je umjetni neuron. Za zadani ulazni vektor $\mathbf{x} = (x_1, \dots, x_n)^T$, izlaz neurona definiramo jednažbom:

$$h = f(\mathbf{w}^T \mathbf{x} + b) \quad (2.33)$$

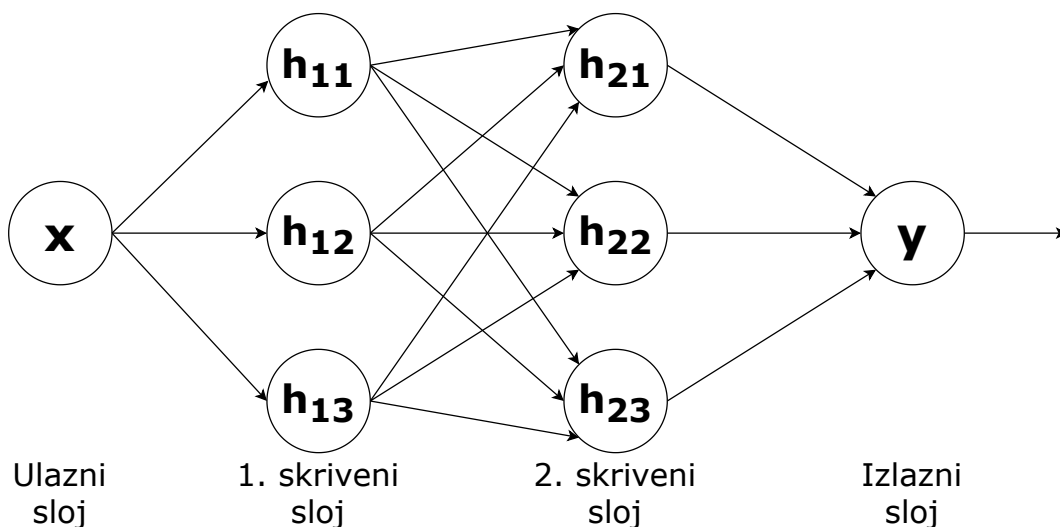
gdje je $\mathbf{w} = (w_1, \dots, w_n)^T$ vektor težina neurona koji određuje doprinos pojedine komponente ulaznog vektora, a skalar b omogućuje dodatni pomak linearne kombinacije ulaza. Aktivacijska funkcija f uvodi nelinearnost u model, čime se omogućuje aproksimacija složenih i nelinearnih odnosa u podacima [10].

Neke od najčešće korištenih aktivacijskih funkcija su sigmoidalna funkcija i hiperbolni tangens, slika 2.4. Obje su nelinearne, monotonno rastuće i kontinuirano diferencijabilne, što je ključno svojstvo u procesu učenja neuronskih mreža metodama koje se temelje na gradijentnom spustu. Sigmoidalna funkcija ima kodomen u intervalu $\langle 0, 1 \rangle$, dok hiperbolni tangens poprima vrijednosti u intervalu $\langle -1, 1 \rangle$, pri čemu obje funkcije imaju karakterističan S-oblik.



Slika 2.4. Primjer aktivacijskih funkcija

Umjetne neurone zatim možemo organizirati u slojeve. Prvi, odnosno ulazni, sloj, zatim proizvoljan broj skrivenih te na kraju izlazni sloj. Time dobivamo osnovnu arhitekturu neuronske mreže, slika 2.5.



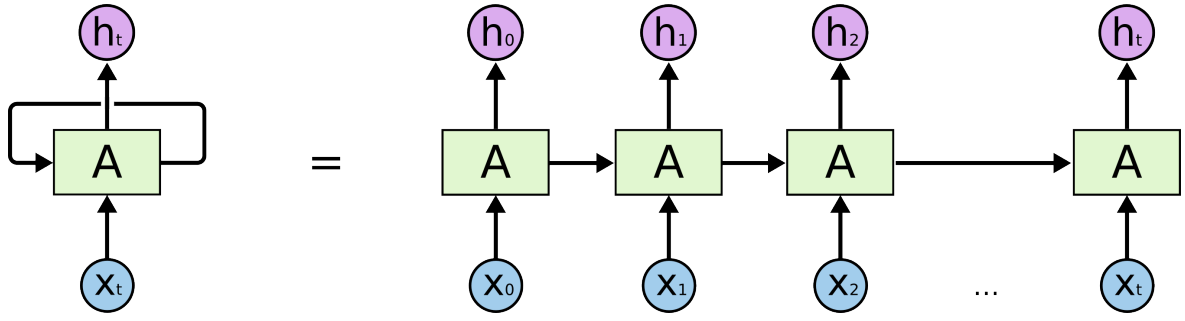
Slika 2.5. Jednostavna neuronska mreža

Među najznačajnijim arhitekturama dubokih modela ističu se duboke unaprijende mreže, konvolucijske neuronske mreže te povratne neuronske mreže. Svaka od navedenih arhitektura prilagođena je specifičnim vrstama podataka i problemima. Odabir odgovarajuće arhitekture ovisi o prirodi i strukturi odabranih podataka.

2.4.1. Povratne neuronske mreže

Dijeljenje parametara jedna je od ranijih ideja u strojnom učenju i statističkom modeliranju, koja omogućuje proširivanje modela na primjere različitih oblika te generalizaciju na podatke varijabilne duljine, osobito u kontekstu sekvencijalnih podataka. Ovakav se princip koristi u skrivenim Markovljevim modelima, gdje se isti skup parametara, poput matrice prijelaza $P(\mathbf{s}_t \mid \mathbf{s}_{t-1})$, koristi u svakom vremenskom koraku. Na taj način model ne ovisi o apsolutnoj poziciji u sekvenci, već isključivo o odnosu između susjednih stanja, što omogućuje modeliranje sekvenci proizvoljne duljine i dijeljenje statističke snage kroz različite vremenske korake. Povratne neuronske mreže (engl. *recurrent neural networks*, *RNN*) preuzimaju i generaliziraju ovu ideju u okviru dubokog učenja. One su prilagođene za obradu sekvencijalnih podataka tako da u svakom vremenskom koraku primjenjuju iste težine i istu transformaciju nad ulazom i prethodnim skrivenim stanjem, pri čemu skriveno stanje \mathbf{h}_t ovisi o trenutnom ulazu \mathbf{x}_t i stanju \mathbf{h}_{t-1} [11].

Slika 2.6. prikazuje dijagram jednostavne povratne neuronske mreže i njezin razmotani (engl. *unrolled*) oblik kroz vremenske korake. Povratnu neuronsku mrežu moguće



Slika 2.6. Jednostavna povratna neuronska mreža [12]

je promatrati kao višestruke kopije iste neuronske mreže, pri čemu svaka kopija prosljeđuje informaciju svojoj sljedbenici u idućem vremenskom koraku. U razmotanom prikazu jasno se vidi da se u svakom koraku primjenjuje ista RNN ćelija A s istim parametrima, dok skriveno stanje predstavlja viđeni dio slijeda koja se prenosi kroz vrijeme. Na taj način model zadržava informaciju o prethodnim ulazima te postupno akumulira kontekst, što mu omogućuje modeliranje ovisnosti unutar sekvence, neovisno o njezinoj duljini [12].

Jednostavni povratni model možemo definirati sljedećim jednadžbama:

$$\mathbf{h}_t = \tanh(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{xh}\mathbf{x}_t + \mathbf{b}_h), \quad (2.34)$$

$$\mathbf{o}_t = \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_o, \quad (2.35)$$

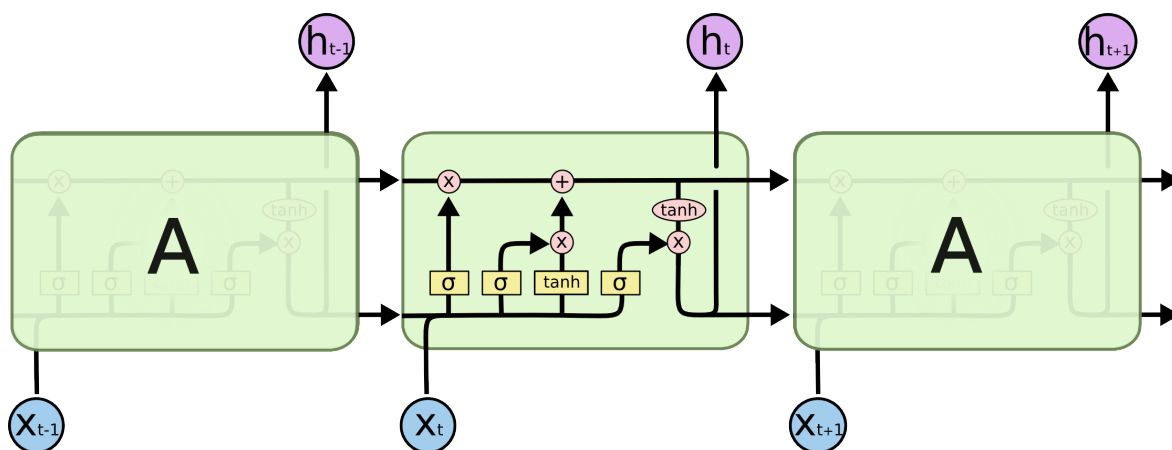
gdje matrica \mathbf{W}_{xh} projicira ulaz u prostor reprezentacije stanja, matrica \mathbf{W}_{hh} modelira evoluciju stanja dok matrica \mathbf{W}_{hy} projicira stanje na prostor predikcija. Vektori \mathbf{b}_h i \mathbf{b}_o omogućuju linearni pomak [13].

Međutim, pri učenju dugoročnih zavisnosti u povratnim neuronskim mrežama pojavljuju se značajni matematički problemi. Gradijenti koji se propagiraju kroz velik broj vremenskih koraka imaju tendenciju eksponencijalnog smanjivanja (problem nestajućeg gradijenta) ili, rjeđe, eksponencijalnog rasta (problem eksplodirajućeg gradijenta), što otežava stabilnu optimizaciju modela. Čak i ako pretpostavimo da su parametri stabilni, utjecaj udaljenih vremenskih koraka postaje eksponencijalno manji u odnosu na nedavne, zbog čega mreža teško uči dugoročne ovisnosti [11].

2.4.2. LSTM mreže

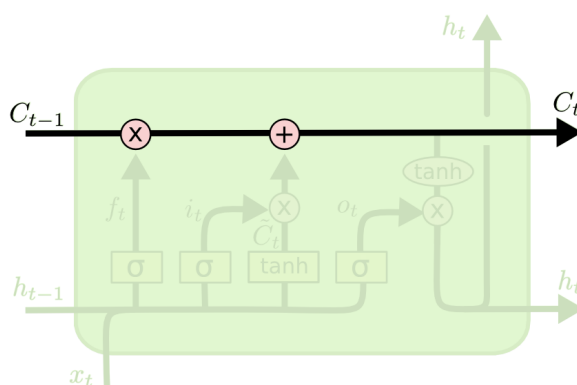
U okviru ovog rada koristit ćemo posebnu vrstu povratnih neuronskih mreža koje nazivamo povratna ćelija s dugoročnom memorijom (engl. *long short-term memory*, *LSTM*). LSTM mreže dizajnirane su kako bi izbjegle probleme nestajućeg i eksplodirajućeg gradijenta te omogućile učenje dugoročnih ovisnosti u sekvencijalnim podacima. Zbog toga se često primjenjuju u modeliranju vremenskih nizova [12].

Razmotana LSTM mreža ima strukturu lančanih ćelija karakterističnu za povratne neuronske mreže, no njene ponavljajuće ćelije imaju složeniju unutarnju arhitekturu. Umjesto jednog neuronskog sloja, LSTM ćelija sastoji se od četiri sloja koji imaju specifičnu interakciju. Ključni element LSTM arhitekture jest stanje ćelije C_t , koje predstavlja



Slika 2.7. Arhitektura LSTM mreže [12]

internu memorijsku komponentu modela. Ono se propagira kroz vremenske korake koristeći samo linearne transformacije, čime se omogućuje stabilniji prijenos informacija i ublažava problem nestajućeg gradijenta. Za razliku od standardnog skrivenog stanja



Slika 2.8. Tok stanja ćelije C_t [12]

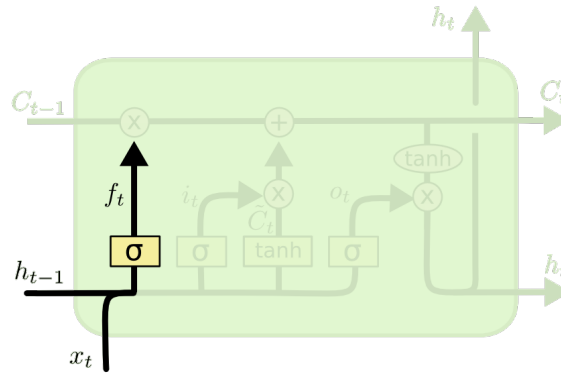
u jednostavnim RNN mrežama, stanje ćelije eksplicitno je regulirano posebnim mehanizmom vrata (engl. *gate*). LSTM arhitektura koristi tri vrste vrata: vrata zaboravljanja, ulazna vrata te izlazna vrata.

Vrata zaboravljanja

Prvi korak u LSTM mreži se odnosi se na određivanje koje informacije iz prethodnog stanja ćelije \mathbf{C}_{t-1} treba izbaciti. Ta se odluka donosi pomoću vrata zaboravljanja, koja kao ulaz primaju prethodno skriveno stanje \mathbf{h}_{t-1} i trenutačni ulaz \mathbf{x}_t . Rezultat je vektor \mathbf{f}_t koji računamo na sljedeći način:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f). \quad (2.36)$$

Pošto vrata zaboravljanja koriste sigmoidu kao aktivacijsku funkciju, $\mathbf{f}_t \in [0, 1]^n$. Elementi stanja \mathbf{C}_t na pozicijama na kojima je vektor \mathbf{f}_t blizu nuli bit će zaboravljeni, dok će se oni na pozicijama na kojima je vektor \mathbf{f}_t blizu jedinici propustiti u iduće stanje.



Slika 2.9. Vrata zaboravljanja [12]

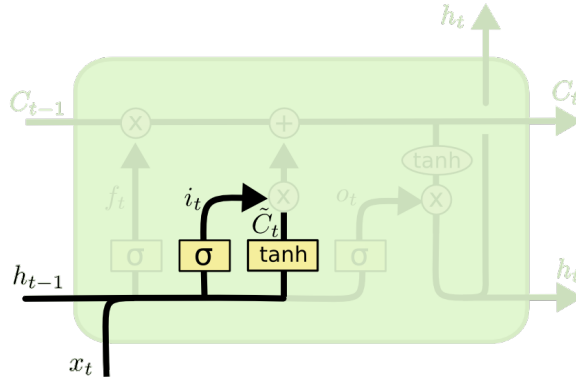
Ulazna vrata

U idućem koraku moramo odrediti koju ćemo informaciju iz ulaza \mathbf{x}_t dodati stanju ćelije. Ovo radimo u dva koraka:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i), \quad (2.37)$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C). \quad (2.38)$$

Vektor \mathbf{i}_t određuje koje informacije je potrebno ažurirati, a sloj koji koristi tangens hiperbolni računa doprinose $\tilde{\mathbf{C}}_t$ iz ulaza \mathbf{x}_t koje ćemo dodati stanju ćelije \mathbf{C}_{t-1} .



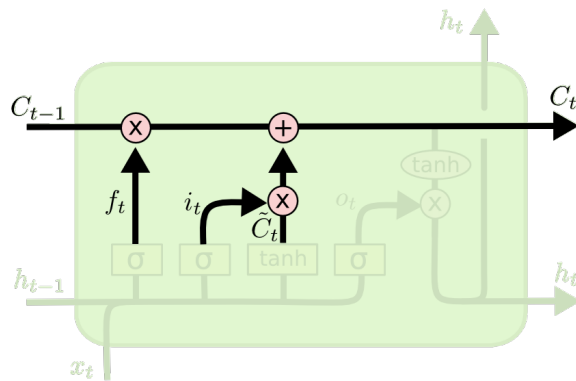
Slika 2.10. Ulazna vrata [12]

Ažuriranje stanja ćelije

Zatim stanje ćelije zatim ažuriramo na sljedeći način:

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t. \quad (2.39)$$

gdje je \odot oznaka za Hadamartov umnožak. Množimo staro stanje s \mathbf{f}_t kako bi zaboravili podatke koje smo odredili ranije. Zatim dodamo umnožak $\mathbf{i}_t \odot \tilde{\mathbf{C}}_t$ koji predstavlja odabrane doprinose ulaza.



Slika 2.11. Ažuriranje stanja ćelije [12]

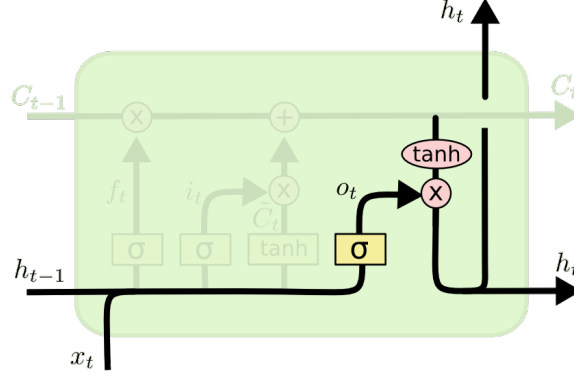
Izlazna vrata

Na kraju, izlaz ćelije će se temeljiti na novom stanju ćelije \mathbf{C}_t . Prvo ćemo pomoću izlaznih vrata \mathbf{o}_t odrediti koje djelove stanja želimo ćemo zadržati na izlazu. Potom ćemo stanje ćelije prvo propustit kroz tangens hiperbolni prije nego ga pomnožimo s izlaznim

vratima:

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o), \quad (2.40)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \quad (2.41)$$



Slika 2.12. Izlazna vrata [12]

2.4.3. Treniranje modela dubokog učenja

Svaki algoritam strojnog učenja možemo rastaviti na tri glavne komponente [14]. Prvu komponentu, odnosno model, možemo definirati kao skup funkcija koje su parametrizirane vektorom parametara θ :

$$\mathcal{H} = \{h(\mathbf{x}; \theta)\}_{\theta}. \quad (2.42)$$

Druga komponenta je funkcija gubitka L (engl. *loss function*) s pripadnom funkcijom pogreške:

$$E(\theta|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N L(y^{(i)}, h(\mathbf{x}^{(i)}; \theta)), \quad (2.43)$$

gdje \mathcal{D} predstavlja skup označenih primjera, a $y^{(i)}$ točnu oznaku primjera $\mathbf{x}^{(i)}$. Funkcija gubitka za dani vektor parametara θ govori koliko se izlaz modela razlikuje od željenog izlaza.

Treća komponenta je optimizacijski postupak:

$$\theta^* = \arg \min_{\theta} E(\theta|\mathcal{D}). \quad (2.44)$$

Cilj optimizacijsko postupka je pronalazak vektora parametara θ^* koji minimiziraju funk-

ciju pogreške. Vektor parametara θ^* jednoznačno određuje funkciju h^* .

Jedan konkretan optimizacijski postupak koji se često koristi kod modela dubokog učenja je gradijentni spust. Ideja gradijentnog spusta je da se, krenuvši od nasumično inicijaliziranog vektora parametara θ , postepeno spuštamo niz površinu funkcije pogreške $E(\theta|\mathcal{D})$ u smjeru suprotnome od gradijenta u točki θ . To ponavljamo dok se ne spustimo u točku u kojoj je gradijent jednak ili jako blizu nuli. Pri tome vektor parametara θ ažuriramo u svakoj iteraciji na sljedeći način:

$$\theta = \theta - \eta \nabla E(\theta|\mathcal{D}) \quad (2.45)$$

gdje je η stopa učenja koja određuje veličinu koraka koje radimo spuštajući se prema minimumu [15].

Međutim fokusiramo li se isključivo na minimiziraju funkciju pogreške pronaći ćemo vektor parametara koji je previše prilagođen na podatke i koji neće dobro raditi na podacima koji nisu bili u skupu za treniranje. Želimo pronaći vektor parametara koji će raditi dobro i za neviđene podatke odnosno koji će dobro generalizirati. Kako bi to postigli, podijelit ćemo skup primjera na skup za treniranje, validaciju i testiranje $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{validation} \cup \mathcal{D}_{test}$. Potupak optimizacije zatim ćemo provoditi samo na skupu za treniranje \mathcal{D}_{train} . Nakon što smo napravili podjelu računamo dvije pogreške. Prva je pogreška funkcije h na skupu za učenje \mathcal{D}_{train} , tj. $E(h|\mathcal{D}_{train})$, a druga pogreška funkcije h na skupu za validaciju $E(h|\mathcal{D}_{validation})$. Postupak optimizacije zaustavit ćemo nakon što smo dosegli minimum pogreške na skupu za validaciju. Nakon toga računamo pogrešku na skupu za testiranje $E(h|\mathcal{D}_{test})$ te će nam ta vrijednost služiti kao mjera performansi dobivene funkcije h . Također pogrešku na skupu za testiranje možemo koristiti kako bi objektivno uspoređivali modele.

3. Implementacija

U ovom poglavlju opisana je programska implementacija predloženog rješenja te alati i biblioteke koje su omogućile učinkovitu izgradnju, treniranje i evaluaciju modela. Implementacija je realizirana u programskom jeziku Python 3 zbog njegove široke primjene u području strojnog učenja i bogatog ekosustava znanstvenih biblioteka. U tablici 3.1. su navedene glavne biblioteke korištene u radu, zajedno s njihovom ulogom u implementaciji.

Tablica 3.1. Korištene Python biblioteke

Biblioteka	Namjena
hydra-core	Upravljanje konfiguracijama i pojednostavljenje podešavanja hiperparametara modela [16]
lightning	Pojednostavljenje i ubrzavanje procesa treniranja modela [17]
matplotlib	Vizualizacija podataka i rezultata [18]
numpy	Rukovanje višedimenzionalnim poljima podataka [19]
pandas	Učitavanje i obrada tabličnih podataka [20]
scipy	Estimacija parametara distribucija i generiranje sintetičkih podataka [21]
torch	Razvoj i treniranje modela strojnog učenja [22]

3.1. Podatci

Kao što je najavljeno u potpoglavlju 2.3.2., model implementiran u ovom radu temelji se na jednofaktorskom modelu povrata. Budući da se u modelu procjenjuju koeficijenti α_i i β_i , na ulazu su potrebni povrati tržišta i dionica iznad bezrizične kamatne stope. Svi korišteni podatci preuzeti su iz baze podataka dostupne na mrežnim stranicama Kenetha

R. Frencha [23].

Tržišni povrati

Podatci o dnevnim aritmetičkim tržišnim povratima i dnevnoj bezrizičnoj kamatnoj stopi preuzeti su iz skupa podataka *Fama/French 3 Factors [Daily]*.

Tržišni povrat definiran je kao vrijednosno otežani (engl. *value weighted*) povrat svih poduzeća iz baze CRSP inkorporiranih u SAD-u i izlistanih na burzama NYSE, AMEX ili NASDAQ, uz zadovoljenje kriterija kvalitete podataka. Takva konstrukcija tržišnog portfelja po svojoj strukturi i obuhvatu vrlo je slična indeksu koji prati SPDR S&P 500 ETF, budući da oba predstavljaju široko diverzificirani presjek američkog tržišta kapitala.

Bezrizična kamatna stopa definirana je kao prinos na jednomjesečni američki trezorski zapis. Do svibnja 2024. godine podatci potječu od Ibbotson Associates, dok se od lipnja 2024. godine koristi referentna stopa iz ICE BofA US 1-Month Treasury Bill Indexa.

Povrati dionica

Podatci o dnevnim aritmetičkim povratima dionica preuzeti su iz skupa *25 Portfolios Formed on Size and Book-to-Market (5 × 5) [Daily]*. Riječ je o 25 portfelja formiranih kao presjek pet portfelja prema veličini poduzeća (engl. *market equity, ME*) i pet portfelja prema omjeru knjigovodstvene i tržišne vrijednosti kapitala (engl. *book-to-market equity, BE/ME*). Portfelji se konstruiraju na kraju lipnja svake godine kao presjeci:

- pet kvintila prema tržišnoj kapitalizaciji ME
- pet kvintila prema omjeru BE/ME

U ovom radu koriste se povrati portfelja, a ne pojedinačnih dionica. Razlog tome je što pojedinačne dionice tijekom vremena ulaze i izlaze s tržišta, što može uzrokovati probleme u konstrukciji uravnoteženih vremenskih nizova.

3.2. Obrada podataka

U ovom potpoglavlju ćemo opisati kako od podataka navedenih u potpoglavlju 3.1. konstruirati značajke za naš model. Prvo nakon što smo učitali podatke o aritmetičkim povratima tržišta, portfelja i bezrizičnu kamatnu stopu, uklanjamo nedostajuće podatke. U našim podacima to su retci u kojima se nalaze vrijednosti -99.99 ili -999. Podatke zatim filtriramo na raspon od 1. 1. 1950. do 28. 11. 2025. uključivo. Odbacujemo podatke prije 1950. godine jer je presjek prvog kvintila po tržišnoj kapitalizaciji i prvog kvintila po omjeru BE/ME bio jako nestabilan¹.

Sljedeći korak je od povrata tržišta i portfelja oduzeti bezrizičnu kamatnu stopu kako bi dobili povrate iznad bezrizične kamatne stope. Neka je R_{Mt} dnevni aritmetički povrata iznad bezrizične kamatne stope tržišta u trenutku t , a R_{it} dnevni aritmetički povrata iznad bezrizične kamatne stope portfelja i u trenutku t . Definirajmo zatim vektore $\mathbf{R}_M = (R_{M1}, \dots, R_{MT})^T$ i $\mathbf{R}_i = (R_{i1}, \dots, R_{iT})^T$ te matricu $\mathbf{R}_{P25} = [\mathbf{R}_1, \dots, \mathbf{R}_{25}]$ oblika $T \times 25$, gdje je T ukupan broj dnevnih povrata u našim podacima.

U idućem koraku vektor \mathbf{R}_M i matricu \mathbf{R}_{P25} potrebno je uzorkovati kliznim prozorom duljine 90 i korakom 30:

$$\mathbf{w}_{Mk} = (R_{Mi}, \dots, R_{M(i+89)})^T \quad i = 30k + 1; \quad 0 \leq k \leq \left\lfloor \frac{T-90}{30} \right\rfloor, k \in \mathbb{Z} \quad (3.1)$$

$$\mathbf{W}_{Pk} = \begin{bmatrix} \mathbf{R}_{P25}^{(i)} \\ \vdots \\ \mathbf{R}_{P25}^{(i+89)} \end{bmatrix} \quad i = 30k + 1; \quad 0 \leq k \leq \left\lfloor \frac{T-90}{30} \right\rfloor, k \in \mathbb{Z} \quad (3.2)$$

gdje je $\mathbf{R}_{P25}^{(i)}$ i -ti redak matrice \mathbf{R}_{P25} . Dobivene prozore zatim dijelimo na način da prvih 60 povrata prozora uzimamo kao povijesne povrate, a posljednjih 30 povrata kao buduće:

$$\mathbf{x}_{Mk} = (w_{Mk1}, \dots, w_{Mk60})^T \quad \mathbf{y}_{Mk} = (w_{Mk61}, \dots, w_{Mk90})^T, \quad (3.3)$$

$$\mathbf{X}_{Pk} = \begin{bmatrix} \mathbf{W}_{Pk}^{(1)} \\ \vdots \\ \mathbf{W}_{Pk}^{(60)} \end{bmatrix} \quad \mathbf{Y}_{Pk} = \begin{bmatrix} \mathbf{W}_{Pk}^{(61)} \\ \vdots \\ \mathbf{W}_{Pk}^{(90)} \end{bmatrix}, \quad (3.4)$$

¹U tim periodima se taj presjek sastojao od svega nekoliko dionica

gdje w_{Mki} i -ti element vektora \mathbf{w}_{Mk} . Primjeti da se podatci, zbog preklapanja prozora, iz \mathbf{R}_M i \mathbf{R}_{P25} ponavljaju u uzorkovanim podacima. To preklapanje je nužno kako bi imali dovoljno uzorka za model. Korak kliznog prozora jednak je veličini budućih prozora kako se barem oni nebi preklapali.

Prije nego što povijesni prozori \mathbf{x}_{Mk} i \mathbf{x}_{Pk} budu spremni za naš model, dodajemo još jednu značajku u obliku interakcije:

$$\mathbf{X}_{Mk} = \mathbf{x}_{Mk} \cdot \mathbf{1}_{25}^T, \quad (3.5)$$

$$\mathbf{X}_{PMk} = \mathbf{X}_{Mk} \odot \mathbf{X}_{Pk}, \quad (3.6)$$

gdje je $\mathbf{1}_n$ vektor jedinica dužine n . Neka je $\mathbf{X}_k = [\mathbf{X}_{Pk}^T, \mathbf{X}_{Mk}^T, \mathbf{X}_{PMk}^T]$ tenzor oblika $25 \times 60 \times 3$. Time smo konstruirali k ulaznih tenzora za naš model.

Jedna od funkcija cilja našeg modela je negativna logaritamska izglednost. Za njezin izračun potrebno nam je očekivanje i kovarijacijska matrica povrata (2.18). S obzirom da naš model pretpostavlja jednofaktorski model povrata (2.32) očekivanje i kovarijacijsku matricu možemo izračunati koristeći formule (2.26) i (2.27). Naš model procjenjuje koeficijente $\hat{\alpha}$ i $\hat{\beta}$, stoga kako bi mogli računati negativnu log izglednost moramo još procijeniti μ_f , σ_f^2 i Ψ iz prozora budućih povrata. S obzirom da je naš faktor predstavljaju povrat tržišta, $\hat{\mu}_{fk}$ i $\hat{\sigma}_{fk}^2$ procjenjujemo iz prozora budućih povrat tržišta \mathbf{y}_{Mk} pomoću izraza (2.9) i (2.10). Kako bi izračunali procjenu kovarijacijske matrice reziduala $\hat{\Psi}$ primjenjujemo formulu (2.16) na rezidualima $\hat{\mathbf{E}}$. Kako bi izračunali $\hat{\mathbf{E}}$ možemo iskoristiti formulu (2.31). Za nju su nam potrebni parametri $\hat{\xi}^T$ koje možemo dobiti primjenom metode najmanjih kvadrata (2.30) na \mathbf{y}_{Mk} i \mathbf{Y}_{Pk} . Nakon što smo dobili parametre $\hat{\xi}_k^T$ možemo procijeniti $\hat{\Psi}_k$. Skup $\mathbf{Y}_k = \{\mathbf{Y}_{Pk}, \mathbf{y}_{Mk}, \hat{\mu}_{fk}, \hat{\sigma}_{fk}^2, \hat{\xi}_k^T, \hat{\Psi}_k\}$ predstavlja ciljne varijable naših k uzorka.

3.3. Sintetički podatci

Osim stvarnih podataka naš model trenirali smo i na sintetičkim podacima. Prednost sintetičkih podataka je to što ih možemo generirati prozirljivo mnogo te su vrijednosti parametara koje pokušavamo procijeniti poznate. Kako bi sintetički podatci mogli pomoći modelu u generalizaciji na stvarne podatke moraju biti što sličniji stvarnim podacima.

Tablica 3.2. Generiranje sintetičkih podataka

Vektor uzoraka	Distribucija	Broj uzoraka
\mathbf{R}_{MS}	$t_5(\hat{\mu}_M, \hat{\sigma}_M^2)$	$N \times 90$
ϵ_S	$\mathcal{N}(\hat{\mu}_\epsilon, \hat{\sigma}_\epsilon^2)$	$N \times 90 \times p$
α_S	$\mathcal{N}(\hat{\mu}_\alpha, \hat{\sigma}_\alpha^2)$	p
β_S	$\mathcal{N}(\hat{\mu}_\beta, \hat{\sigma}_\beta^2)$	p

Za simulaciju tržišnih povrata odabralismo studentovu t -distribuciju s pet stupnjeva slobode. Za procjenu μ_M i σ_M^2 koristili smo procjenu najveće izglednosti koju nudi biblioteka `scipy` na podacima \mathbf{R}_M . Primjenjivanjem metode najmanjih kvadrata (2.30) na \mathbf{W}_{Mk} i \mathbf{W}_{Pk} dobivamo k parametara $\hat{\xi}_k$. To nam omogućuje da uz formulu (2.31) dobijemo k reziduala $\hat{\mathbf{E}}_k$. Iz parametra $\hat{\xi}_k$ su lako dostupni $\hat{\alpha}_k$ i $\hat{\beta}_k$. Potom sve elemente $\hat{\mathbf{E}}_k$, $\hat{\alpha}_k$ i $\hat{\beta}_k$ promatramo kao realizacije triju normalnih distribucija $\{\epsilon_i\}_{i=1}^{90 \times 25 \times k}$, $\{\alpha_i\}_{i=1}^{25 \times k}$ i $\{\beta_i\}_{i=1}^{25 \times k}$. Primjenom formula (2.9) i (2.10) dobivamo procjene $\hat{\mu}_\epsilon$, $\hat{\sigma}_\epsilon^2$, $\hat{\mu}_\alpha$, $\hat{\sigma}_\alpha^2$, $\hat{\mu}_\beta$ i $\hat{\sigma}_\beta^2$. Sintetičke podatke zatim generiramo kako je opisano u tablici 3.2., gdje je N željeni broj uzoraka, a p broj simuliranih portfelja. Nakon toga vektor ϵ_S preoblikujemo u matricu \mathbf{E}_S oblika $(N \times 90) \times p$. Generirani vektor \mathbf{R}_{MS} predstavlja povrate tržišta \mathbf{R}_M , a simulirane povrate portfelja \mathbf{R}_{P25} dobivamo na sljedeći način:

$$\mathbf{A}_S = \mathbf{1}_{N \times 90} \cdot \alpha_S^T \quad \text{matrica } (N \times 90) \times p, \quad (3.7)$$

$$\mathbf{R}_{PS} = \mathbf{A}_S + \mathbf{R}_{MS} \beta_S^T + \mathbf{E}_S \quad (3.8)$$

Nakon što imamo povrate tržišta \mathbf{R}_{MS} i portfelja \mathbf{R}_{PS} možemo ponoviti korake iz poglavlja 3.2. kako bi generirali ulazne tenzore i ciljne varijable. Postoje samo dvije razlike. Kod sintetičkih podataka korak kliznog prozora postavljamo na 90 jer nema razloga da nam se sintetički podatci preklapaju. Kod procjenu kovarijacijske matrice reziduala $\hat{\Psi}$ nema potrebe da procjenjujemo parametre $\hat{\xi}$ metodom najmanjih kvadrata jer su nam oni već poznati $\xi = [\alpha_S, \beta_S]$.

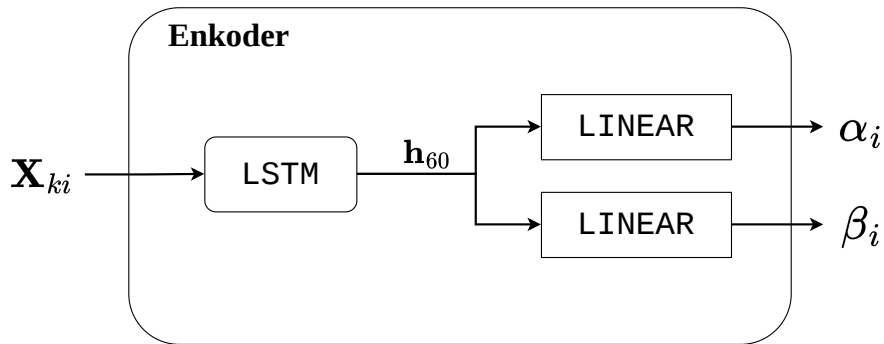
3.4. Model dubokog učenja

TODO Razradit ovaj paragraf

Naš model sastoji se od enkodera i nekoliko različitih dekodera prilagođenih različitim funkcijama gubitka. S obzirom na ovu arhitekturu naš model smatramo generativnim modelom. Enkoder naučiti distribuciju budućih povrata za dane povijesne te iz toga pokušava procijeniti parametre jednofaktorskog modela. Nakon što imamo parametre možemo generirati razne prognoze i testirati različite scenarije.

Enkoder

Glavna komponenta našeg modela je enkoder koji se sastoji od jedne LSTM jedinice i dva linearna sloja. Neka je \mathbf{X}_{ki} i -ta od p matrica iz tenzora \mathbf{X}_k gdje je p broj portfelja i prva dimenzija tenzora \mathbf{X}_k . Matrica \mathbf{X}_{ki} ima oblik 60×3 i sadrži 60 povrata i -tog portfelja, tržišta te njihov produkt za k -ti prozor podataka. Enkoder na ulazu prima ta tri vremenska niza i prosljeđuje ih LSTM jedinici. Potom uzimamo posljednje skriveno stanje LSTM jedinice \mathbf{h}_{60} i stavljamo ga na ulaz dva zasebna linearna sloja. Zadaća linearnih slojeva je procijeniti koeficijente α_i i β_i jednofaktorskog modela koji najbolje odgovarati budućem prozoru povrata iz posljednjeg skrivenog stanja \mathbf{h}_{60} . Slika 3.1. prikazuje arhitekturu enkodera. Enkoderu također možemo primjeniti na čitav tenzor \mathbf{X}_k čime dobivamo vektore α i β , odnosno procjenu koeficijenata α_i i β_i za svih p portfelja.



Slika 3.1. Enkoder

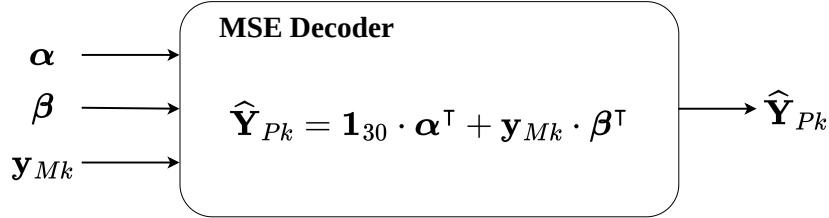
Dekoderi i funkcije gubitka

U ovom radu ispitali smo tri različite funkcije gubitka od kojih je svaka zahtjevala zaseban dekoder. Prva funkcija gubitka je srednja kvadratna pogreška (engl. *mean squared error*, *MSE*). Srednju kvadratnu pogrešku našeg modela računat ćemo na sljedeći način:

$$\text{MSE}(\hat{\mathbf{Y}}_{Pk}, \mathbf{Y}_{Pk}) = \frac{1}{p \times 30} \sum_{i=1}^p \sum_{j=1}^{30} \left(\mathbf{Y}_{Pk}^{(j,i)} - \hat{\mathbf{Y}}_{Pk}^{(j,i)} \right)^2, \quad (3.9)$$

gdje je $\mathbf{Y}_{Pk}^{(j,i)}$ element matrice \mathbf{Y}_{Pk} u i -tom stupcu i j -tom retku. Kako bi mogli izračunati srednju kvadratnu pogrešku prvi dekodler će izračunati $\hat{\mathbf{Y}}_{Pk}$ na sljedeći način:

$$\hat{\mathbf{Y}}_{Pk} = \mathbf{1}_{30} \cdot \boldsymbol{\alpha}^\top + \mathbf{y}_{Mk} \cdot \boldsymbol{\beta}^\top. \quad (3.10)$$

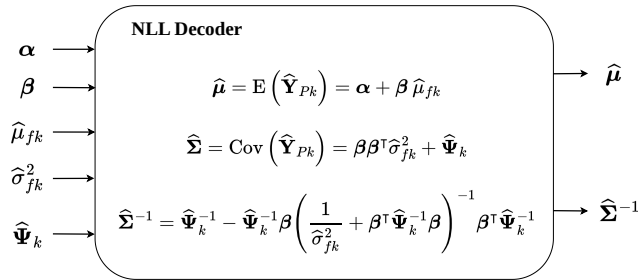


Slika 3.2. MSE Dekoder

Druga funkcija gubitka je negativna logaritamsku izglednost (engl. *negative log likelihood*, *NLL*). Negativnu logaritamsku izglednost našeg modela računamo na sljedeći način:

$$\text{NLL}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}^{-1}, \mathbf{Y}_{Pk}) = -\frac{1}{30} \ln \mathcal{L}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}^{-1}; \mathbf{Y}_{Pk}). \quad (3.11)$$

Drugi dekodler računa procjene $\hat{\boldsymbol{\mu}}$ i $\hat{\boldsymbol{\Sigma}}^{-1}$ primjenjujući formule (2.26) i (2.27) na $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ i podatke iz skupa \mathbf{Y}_k . Kod računanja inverza $\hat{\boldsymbol{\Sigma}}^{-1}$ poslužili smo se Woodburyjevom lemom o invertiranju matrica [24].

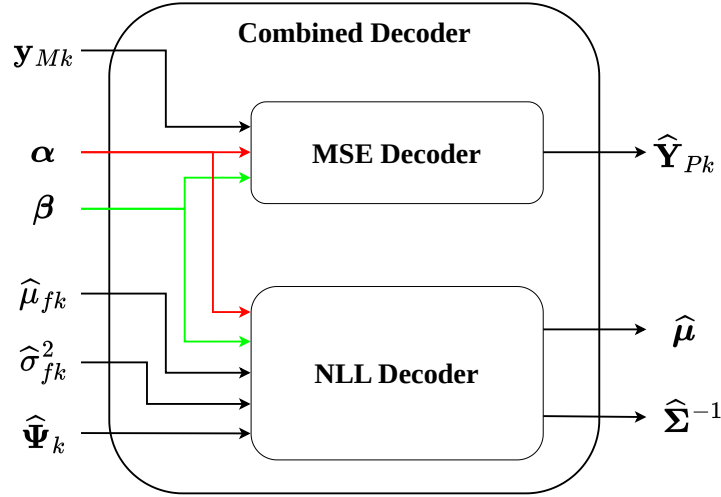


Slika 3.3. NLL Dekoder

Treća funkcija gubitka je zapravo samo kombinacija srednje kvadratne pogreške i negativne logaritamske izglednosti te je računamo na idući način:

$$\text{COMBINED}(\hat{\mathbf{Y}}_{Pk}, \mathbf{Y}_{Pk}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}^{-1}) = \text{NLL}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}^{-1}, \mathbf{Y}_{Pk}) + \zeta \times \text{MSE}(\hat{\mathbf{Y}}_{Pk}, \mathbf{Y}_{Pk}), \quad (3.12)$$

gdje koeficijent ζ određuje doprinos srednje kvadratne pogreške. Zbog tog je treći dekodler zapravo samo objedinjenje prvog i drugog dekodera u jedan veći dekodler.



Slika 3.4. Kombinirani Dekoder

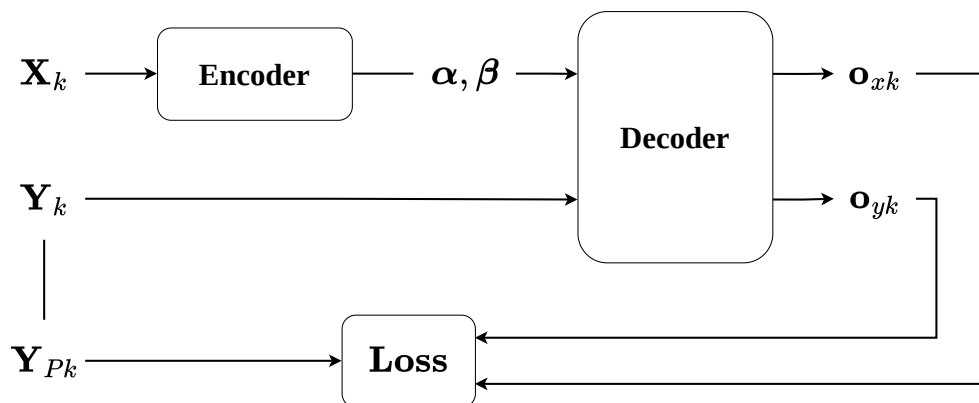
Računanje ukupnog gubitka modela

Sva tri dekodera našeg modela na ulazu prima podatke iz skupa \mathbf{Y}_k i vektore procjenjenih koeficijenata α i β koje dobivamo na izlazu enkodera. Osim izlaza enkodera, skup \mathbf{Y}_k također sadrži vektore parametara α_k i β_k koji se nalaze u matrici parametara $\hat{\xi}_k^\top$. Ti parametri procjenjeni su metodom najmanjih kvadrata ne budućem prozoru povrata. Naš cilj je da model daje jednako kvalitetne procjene parametara kao i metoda najmanjih kvadrata ne budućem prozoru povrata, ali koristeći samo povijesni prozor povrata. Stoga ćemo gubitak izračunat na parametrima α_k i β_k smatrati osnovnom razinom gubitka.

Neka je \mathbf{o}_{xk} izlaz dekodera za ulazne vektore α i β dobivene primjenom enkodera na tenzor \mathbf{X}_k , a \mathbf{o}_{yk} izlaz dekodera za ulazne vektore α_k i β_k dobivene iz skupa \mathbf{Y}_k . Ukupan gubitak modela koji koristimo u postupku optimizacije za primjer k tada računamo na sljedeći način:

$$\text{LOSS}_{TOTAL}(\mathbf{o}_{xk}, \mathbf{o}_{yk}, \mathbf{Y}_{Pk}) = \text{LOSS}(\mathbf{o}_{xk}, \mathbf{Y}_{Pk}) - \text{LOSS}(\mathbf{o}_{yk}, \mathbf{Y}_{Pk}) \quad (3.13)$$

Općenit oblik našeg modela tada možemo prikazati slikom 3.5.



Slika 3.5. Općeniti oblik modela

4. Rezultati

TODO Rezultati i rasprava

5. Zaključak

TODO Zaključak

Literatura

- [1] S. Begušić i Z. Kostanjčar, “Financijska tržišta i instrumenti”, Prezentacija s kolegija Analitika financija na FER-u, 2025.
- [2] J. Y. Campbell, A. W. Lo, i A. C. MacKinlay, *The Econometrics of Financial Markets*, 2. izd. Princeton University Press, 1997.
- [3] R. S. Tsay, *Analysis of financial time series*, 3. izd. John Wiley & Sons, Inc., 2010.
- [4] N. L. Johnson, S. Kotz, i N. Balakrishnan, *Continuous Univariate Distributions*, 2. izd. John Wiley & Sons, Inc., 1994., sv. 1.
- [5] —, *Continuous Univariate Distributions*, 2. izd. John Wiley & Sons, Inc., 1994., sv. 2.
- [6] R. Cont, “Empirical properties of asset returns: stylized facts and statistical issues”, *Quantitative Finance*, sv. 1, br. 2, str. 223–236, 2001. <https://doi.org/10.1080/713665670>
- [7] M. Taboga, “Multivariate normal distribution - maximum likelihood estimation”, Lectures on probability theory and mathematical statistics. Kindle Direct Publishing. Online appendix., 2021., <https://www.statlect.com/fundamentals-of-statistics/multivariate-normal-distribution-maximum-likelihood>, Pristupljeno 13. 2. 2026.
- [8] S. Begušić i Z. Kostanjčar, “Modeli povrata imovina”, Prezentacija s kolegija Analitika financija na FER-u, 2025.
- [9] R. A. Johnson i D. W. Wichern, *Applied multivariate statistical analysis*, 5. izd. Pearson Education, 2002.

- [10] J. Krapac i S. Šegivć, “Unaprijedni duboki modeli”, Prezentacija s kolegija Duboko učenje 1 na FER-u, 2025.
- [11] I. Goodfellow, Y. Bengio, i A. Courville, *Deep Learning*. MIT Press, 2016., <http://www.deeplearningbook.org>.
- [12] C. Olah, “Understanding lstm networks”, 2015., <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, Pristupljeno 13. 2. 2026.
- [13] M. Tutek, P. Bevandić, J. Šarić, i S. Šegvić, “Jednostavni povratni modeli”, Prezentacija s kolegija Duboko učenje 1 na FER-u, 2025.
- [14] J. Šnajder, “Osnovni koncepti”, Skripta s kolegija Strojno učenje 1 na FER-u, 2022.
- [15] —, “Logistička regresija”, Skripta s kolegija Strojno učenje 1 na FER-u, 2022.
- [16] Hydra, <https://hydra.cc/>, verzija 1.3.2.
- [17] Lightning, <https://lightning.ai/docs/pytorch/latest/>, verzija 2.6.0.
- [18] Matplotlib, <https://matplotlib.org/>, verzija 3.10.8.
- [19] Numpy, <https://numpy.org/>, verzija 2.4.1.
- [20] Pandas, <https://pandas.pydata.org/>, verzija 3.0.0.
- [21] SciPy, <https://scipy.org/>, verzija 1.17.0.
- [22] PyTorch, <https://pytorch.org/>, verzija 2.10.0.
- [23] E. F. Fama i K. R. French, “U.s. research returns data”, https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html, Pristupljeno 13. 2. 2026.
- [24] M. Taboga, “Matrix inversion lemmas”, Lectures on matrix algebra, 2021., <https://www.statlect.com/matrix-algebra/matrix-inversion-lemmas>, Pristupljeno 13. 2. 2026.

Sažetak

TBD

Marko Miljković

Unesite sažetak na hrvatskom.

Ključne riječi: prva ključna riječ; druga ključna riječ; treća ključna riječ

Abstract

TBD

Marko Miljković

Enter the abstract in English.

Keywords: the first keyword; the second keyword; the third keyword