Tema 6: Globoko učenje za analizo vplivov na okolje; predmet: Strojno učenje za okoljske aplikacije

Učni izidi

Po zaključku te enote bodo študenti znali:

- Razložiti koncept in vlogo nevronskih mrež v okoljskih aplikacijah.
- Razložiti temeljne zamisli globokega učenja (DL) ter način zgradbe in učenja nevronskih mrež.
- Opisati uporabo konvolucijskih in rekurentnih nevronskih mrež pri analizi vplivov na okolje.
- Uporabiti osnovni model globokega učenja za analizo okoljskih podatkov (npr. onesnaženosti, pokrovnosti tal ali temperaturnih trendov).
- Interpretirati rezultate modelov globokega učenja in razpravljati o njihovi vlogi pri podpori trajnostnemu odločanju.
- Razviti preproste prototipe ali analize scenarijev, ki prikazujejo uporabo globokega učenja za trajnost.
- Razpravljati o tem, kako lahko DL pomaga odločevalcem pri ocenjevanju ekoloških tveganj.

Uvod

Globoko učenje (DL) je veja strojnega učenja, ki uporablja umetne nevronske mreže za učenje kompleksnih vzorcev v podatkih. V nasprotju s tradicionalnimi algoritmi lahko modeli globokega učenja samodejno izluščijo pomembne značilke, zato so še posebej uporabni za velike in kompleksne zbirke okoljskih podatkov. To predavanje predstavlja temeljne zamisli globokega učenja in prikazuje, kako te metode podpirajo prilagajanje podnebnim spremembam, varstvo okolja in trajnostno odločanje.

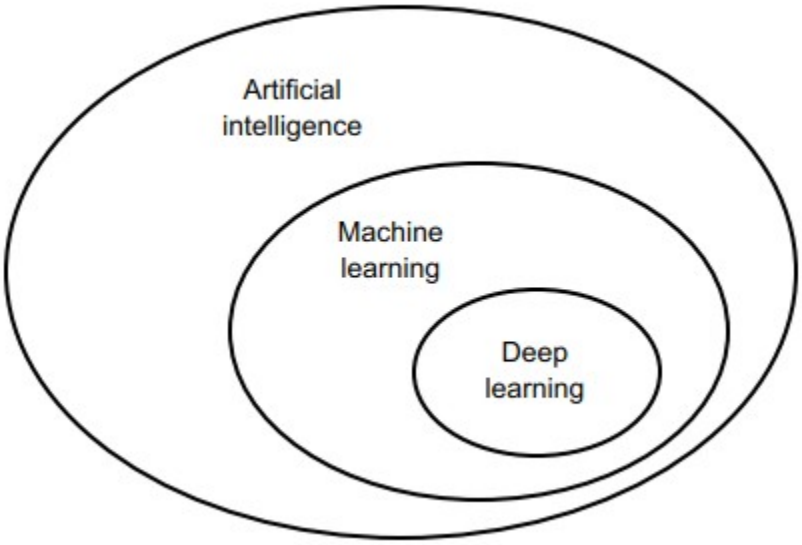
Opredelitev in razlikovanje: strojno učenje v primerjavi z globokim učenjem

Umetna inteligenca se je rodila v petdesetih letih 20. stoletja, ko se je peščica pionirjev z nastajajočega področja računalništva začela spraševati, ali bi lahko računalnike pripravili do tega, da bi »razmišljali« – vprašanje, katerega posledice raziskujemo še danes [1]. Jedrnato

lahko UI opišemo kot prizadevanje za avtomatizacijo intelektualnih nalog, ki jih običajno opravljajo ljudje. UI je tako splošno področje, ki zajema strojno in globoko učenje, pa tudi številne druge pristope, ki morda sploh ne vključujejo učenja. Razmerje med umetno inteligenco, strojnim učenjem in globokim učenjem prikazujejo trije vgnezdeni ovali na sliki 6.1.

Strojno učenje je način programiranja računalnikov, pri katerem se lahko ti učijo iz podatkov in izboljšujejo svoje delovanje, ne da bi jim za vsako situacijo podali natančno pravilo. Pri klasičnem strojnem učenju ljudje navadno izberejo, katere vhodne značilnosti so pomembne, na primer »povprečna temperatura«, »količina padavin« ali »vlažnost tal«, nato pa algoritem naučijo, da na podlagi teh značilk napoveduje ali sprejema odločitve. Ta pristop je učinkovit, kadar je podatkov razmeroma malo, so strukturirani v preglednicah ter je pomembno, da modeli ostanejo preprosti in jih je mogoče zlahka razložiti.

Globoko učenje je posebna vrsta strojnega učenja, ki uporablja večplastne umetne nevronske mreže za neposredno učenje iz neobdelanih podatkov, kot so slike, zvok ali dolge časovne vrste. Namesto da bi ljudje ročno načrtovali značilke, se globoka nevronska mreža v svojih plasteh korak za korakom samodejno nauči uporabnih vzorcev, na primer od slikovnih pik prek oblik do predmetov na sliki. Globoko učenje običajno zahteva veliko več podatkov in računske moči, vendar je zelo uspešno pri reševanju kompleksnih nalog, kot so prepoznavanje slik, prepoznavanje govora in razumevanje naravnega jezika.



Slika 6.1 Ravni umetne inteligence

Prevod napisov na sliki: Umetna inteligenca; strojno učenje; globoko učenje.

Ključna razlika je v tem, da je strojno učenje širša družina metod, pri katerih ljudje navadno določijo značilke, modeli pa so pogosto manjši in jih je lažje interpretirati, medtem ko je globoko učenje podmnožica, ki temelji na globokih nevronskih mrežah, da se značilk in vzorcev samodejno nauči iz velikih količin podatkov. V praksi se strojno učenje pogosto uporablja za strukturirane okoljske podatke v preglednicah ali podatkovnih bazah, globoko učenje pa je primernejše za delo s satelitskimi posnetki, mrežami senzorjev ali drugimi visokorazsežnimi signali. Pri tem predmetu naj si študenti zapomnijo preprosto pravilo: vse globoko učenje je strojno učenje, ni pa vse strojno učenje globoko učenje.

Eksplozija okoljskih podatkov: sateliti, droni, senzorji interneta stvari, podnebni modeli

Sodobna okoljska znanost je močno presegla preproste ročne meritve; danes imamo globalno omrežje digitalnih »oči« in »ušes«, ki spremljajo Zemljo. Sateliti vsak dan zajemajo visokoločljive posnetke planeta ter spremljajo spremembe, kot sta taljenje ledu ali rast mest, medtem ko droni tridimenzionalno kartirajo gozdove in obale. Hkrati milijoni senzorjev interneta stvari (IoT), nameščenih v oceanih, tleh in mestih, neprekinjeno posredujejo podatke o temperaturi, onesnaženosti in vlažnosti. Tako nastaja ogromna, neurejena zbirka informacij –

pogosto imenovanih »nestrukturirani podatki« –, ki je veliko preobsežna, da bi jo ljudje lahko analizirali ročno.

Poleg opazovanj resničnega sveta se znanstveniki zanašajo na kompleksne podnebne modele, ki tečejo na superračunalnikih in simulirajo vremenske vzorce za desetletja v prihodnost. Ko te simulacije združimo s satelitskimi posnetki in tokovi podatkov senzorjev, dobimo podatkovne zbirke z milijoni spremenljivk oziroma »visokorazsežne« podatke. Tradicionalne metode strojnega učenja imajo pri tem pogosto težave, saj morajo ljudje podatke najprej ročno poenostaviti (inženirstvo značilk). Zaradi same količine podatkov, ki jih ustvarjamo danes, je ročno oblikovanje pravil za vsak satelitski posnetek ali odčitek senzorja nemogoče.

Globoko učenje postane nujno. V nasprotju s starejšimi algoritmi, ki se po določeni točki prenehajo izboljševati, modeli globokega učenja dejansko delujejo bolje, ko dobijo več podatkov. Edinstveno so sposobni obdelati to »eksplozijo podatkov« – s konvolucijskimi nevronskimi mrežami (CNN) »vidijo« vzorce na satelitskih posnetkih, z rekurentnimi nevronskimi mrežami (RNN) pa si »zapomnijo« trende v dnevnikih podnebnih senzorjev. Globoko učenje z avtomatizacijo analize teh kompleksnih podatkov spreminja poplavo neobdelanih informacij v uporabna spoznanja za varovanje našega planeta.

Zakaj je globoko učenje pomembno: obravnavanje kompleksnih, nelinearnih in visokorazsežnih podatkov

Globoko učenje je bistveno za okoljsko znanost, ker narava redko sledi ravni črti; ekološka razmerja so namesto tega izrazito **nelinearna**, kar pomeni, da lahko majhna sprememba enega dejavnika (na primer temperature) sproži nesorazmerno velike ali kompleksne učinke (na primer nevihto). Medtem ko tradicionalni algoritmi težko »rišejo« te kompleksne krivulje, modeli globokega učenja z aktivacijskimi funkcijami učinkovito približajo vsako zvezno, ukrivljeno razmerje v naravi. Poleg tega so okoljski podatki pogosto **visokorazsežni** in sestavljeni iz obsežnih vhodnih podatkov, kot so satelitski posnetki z milijoni slikovnih pik ali sočasni odčitki tisočev vremenskih senzorjev. Starejše metode pogosto preobremeni že sama količina informacij – ta problem je znan kot »prekletstvo razsežnosti« –, arhitekture globokega učenja pa so posebej zasnovane za obravnavo tako obsežnih, nestrukturiranih podatkovnih zbirk. Samodejno ločijo šum od pomembnih vzorcev, ne da bi morali ljudje podatke predhodno poenostaviti. Primeri uporabe DL na področju trajnosti vključujejo:

- Kartiranje krčenja gozdov. Konvolucijske nevronske mreže (CNN) analizirajo satelitske posnetke območij, kot je Amazonija, in samodejno zaznajo, kje je bil gozd posekan ali degradiran, s čimer oblastem pomagajo hitreje ukrepati in načrtovati varstvene ukrepe.
- Napovedovanje poplav. Rekurentni ali hibridni modeli globokega učenja se učijo iz preteklih podatkov o padavinah, vodostajih rek in vlažnosti tal ter napovedujejo, kdaj in kje bodo reke verjetno prestopile bregove. Tako imajo skupnosti več časa za pripravo in zmanjšanje škode.
- Razvrščanje koralnih grebenov. Modeli globokega učenja, naučeni na tisočih podvodnih fotografij, lahko prepoznajo različne vrste koral ter zaznajo beljenje ali bolezni, zato morskim biologom omogočajo, da zdravje grebenov spremljajo veliko hitreje kot z ročnimi pregledi.
- Napovedovanje emisij. Modeli globokega učenja lahko uporabijo podatke o porabi energije, prometu in industrijski dejavnosti za oceno dnevnih ali prihodnjih emisij ogljikovega dioksida ter tako vladam in podjetjem pomagajo spremljati napredek pri doseganju podnebnih ciljev.

Osnove globokih nevronskih mrež

Ta modul postavlja teoretični in praktični okvir **globokega učenja (DL)**, podpodročja umetne inteligence, ki je korenito spremenilo način interpretiranja kompleksnih podatkovnih zbirk. DL pomeni premik od tradicionalnega statističnega modeliranja k samodejnemu inženirstvu značilk.

Tradicionalne statistične metode, kot je linearna regresija, so močno odvisne od ročnega izbora značilk – raziskovalci morajo določiti in predhodno obdelati posamezne spremenljivke, da bi ugotovili njihov vpliv –, globoke nevronske mreže (DNN) pa so zasnovane tako, da se teh predstavitev naučijo samostojno. DNN prek zaporedja skritih plasti razčleni neobdelane podatke na več ravni abstrakcije.

- **Začetne plasti:** zajamejo preproste vzorce ali značilke »nizke ravni«.
- **Globlje plasti:** združujejo te vzorce, da prepoznajo zapletene strukture visoke ravni v podatkih.

Ta sposobnost hierarhičnega učenja je še posebej pomembna za **okoljsko analitiko**. Naravne sisteme redko določajo preprosta, aditivna razmerja; opredeljujejo jih spremenljivke – kot so vlažnost, atmosferski tlak in trdni delci –, ki medsebojno delujejo nelinearno in zelo kompleksno.

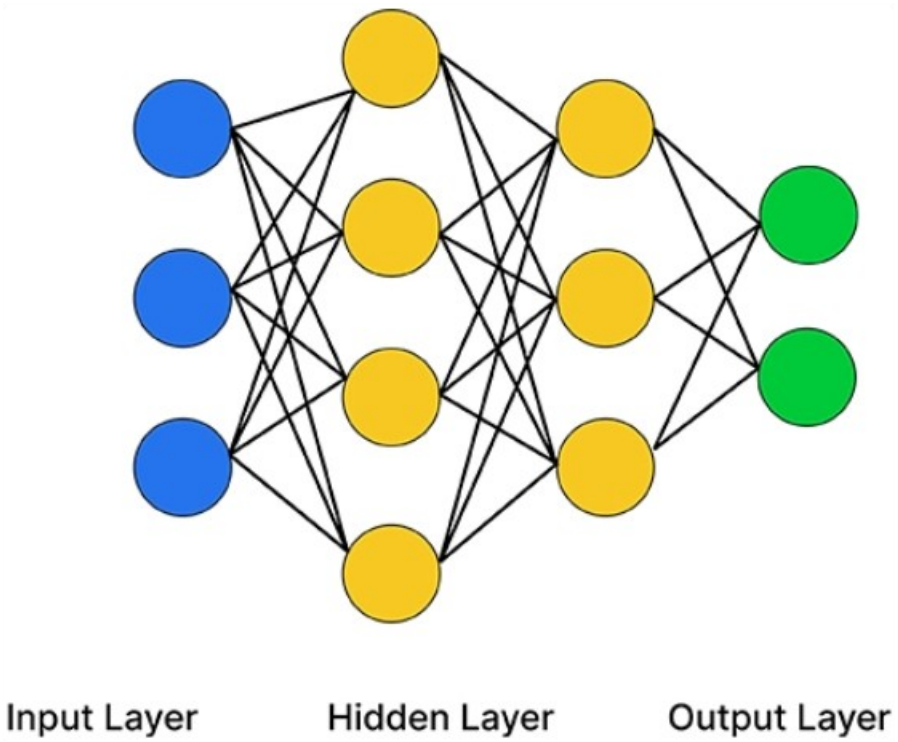
Tradicionalni modeli imajo pogosto težave s temi »kaotičnimi« medsebojnimi vplivi, DNN pa se odlikujejo pri:

- **Prepoznavanju vzorcev:** odkrivanju komaj opaznih sprememb v podnebnih podatkih, ki nastopijo pred ekstremnimi vremenskimi dogodki.
- **Večrazsežna korelacija:** kartiranje tega, kako se različni okoljski dejavniki (na primer vlažnost tal in hitrost vetra) združujejo ter vplivajo na lokalne ekosisteme.
- **Natančnosti napovedovanja:** zagotavljanju zanesljivejših napovedi z zajemanjem nelinearnega »šuma«, ki bi ga preprostejši modeli lahko zavrgli kot osamelce.

Z obvladovanjem teh temeljev lahko raziskovalci presežejo zgolj opazovanje podatkov in začnejo graditi napovedna orodja, ki so kos obsegu in spremenljivosti našega spreminjajočega se planeta.

Arhitektura nevronskih mrež: koncept plasti

Nevronsko mrežo lahko razumemo kot matematično funkcijo, ki neko množico vhodnih vrednosti preslika v izhodne vrednosti.

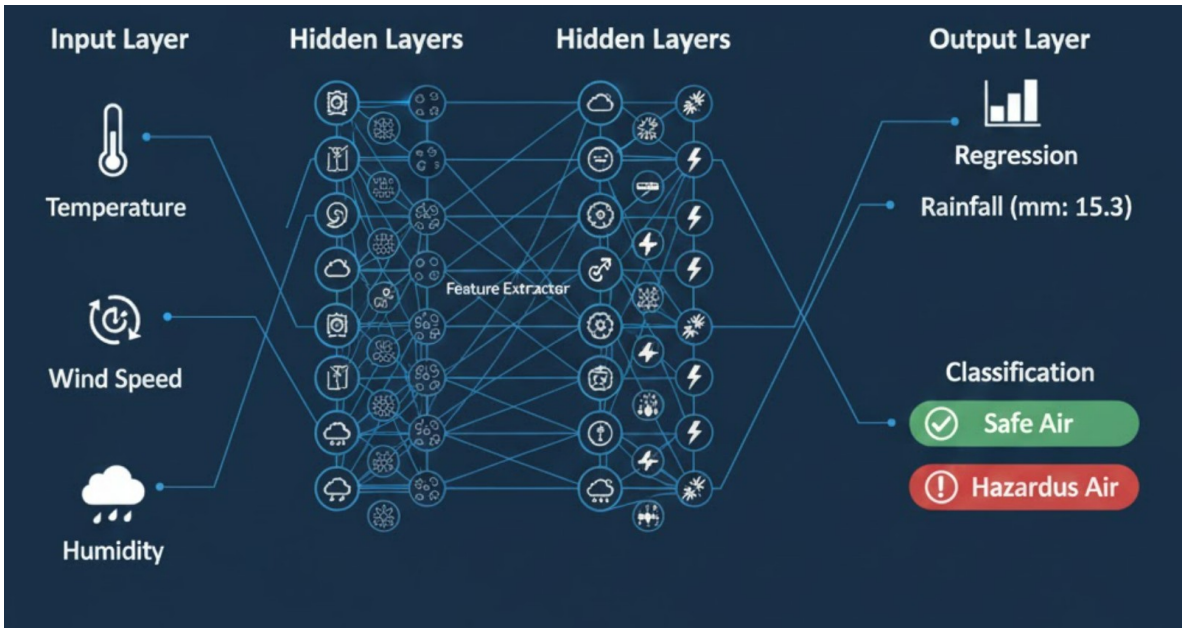


Slika 6.2 Arhitektura nevronskih mrež

Prevod napisov na sliki: Vhodna plast; skrita plast; izhodna plast.

Globoka mreža s širjenjem naprej [1] oziroma večplastni perceptron (MLP) je sestavljena iz treh vrst plasti (slika 6.2):

- **Vhodna plast:**
  Ta plast sprejema neobdelane okoljske podatke. Ne izvaja nobenih izračunov. Število nevronov v tej plasti natančno ustreza številu značilk v podatkovni zbirki.
  - *Okoljski kontekst:* Če analiziramo kakovost zraka, lahko vhodni nevroni predstavljajo posamezne odčitke senzorjev, na primer: $[x_1: \text{Temperature}, x_2: \text{Wind Speed}, x_3: \text{Humidity}]$.
- **Skrite plasti:**
  Te plasti so umeščene med vhodno in izhodno plast. Imenujemo jih »skrite«, ker njihovih vrednosti v učnih podatkih ne opazujemo.
  - **Heaton** pojasnjuje, da mreža ob prehajanju podatkov skozi te plasti izračunava utežene vsote vhodov. »Globoko« učenje preprosto pomeni, da ima mreža več kot eno skrito plast.
  - *Funkcija:* Te plasti delujejo kot izluščevalniki značilk. Zgodnje plasti lahko zaznajo preproste vzorce (npr. dnevne temperaturne cikle), globlje plasti pa jih združujejo, da zaznajo kompleksne pojave (npr. posebne razmere pred vročinskim valom).
- **Izhodna plast:**
  Končna plast ustvari napoved (slika 6.3).
  - *Regresija:* En sam nevron, ki na izhodu poda zvezno vrednost (npr. napovedane padavine v mm).
  - *Razvrščanje:* Več nevronov, ki predstavljajo verjetnosti različnih razredov (npr. razred 0: »varen zrak«, razred 1: »nevaren zrak«).



Slika 6.3 Plasti nevronske mreže

Prevod napisov na sliki: Vhodna plast: temperatura, hitrost vetra, vlažnost; skrite plasti in izluščevalnik značilk; izhodna plast: regresija (padavine) ali klasifikacija (varen/nevaren zrak).

Aktivacijske funkcije: uvedba nelinearnosti

Če bi bila nevronska mreža sestavljena samo iz uteženih vsot in prištevanja odmikov, bi ne glede na število plasti ostala linearni model – pravzaprav velikanska regresijska enačba. Kompozicija linearnih funkcij je namreč tudi sama linearna funkcija, zato mreža nikoli ne bi mogla dojeti zapletenih, »nazobčanih« ali ukrivljenih razmerij v podatkih iz resničnega sveta.

Za modeliranje kompleksnih okoljskih pojavov – kot so **nelinearne stopnje rasti bakterij** ali **kaotični prag nevihtne celice** – moramo pri vsakem nevronu uvesti matematičnega »vratarja«. To je vloga **aktivacijske funkcije** (slika 6.4).

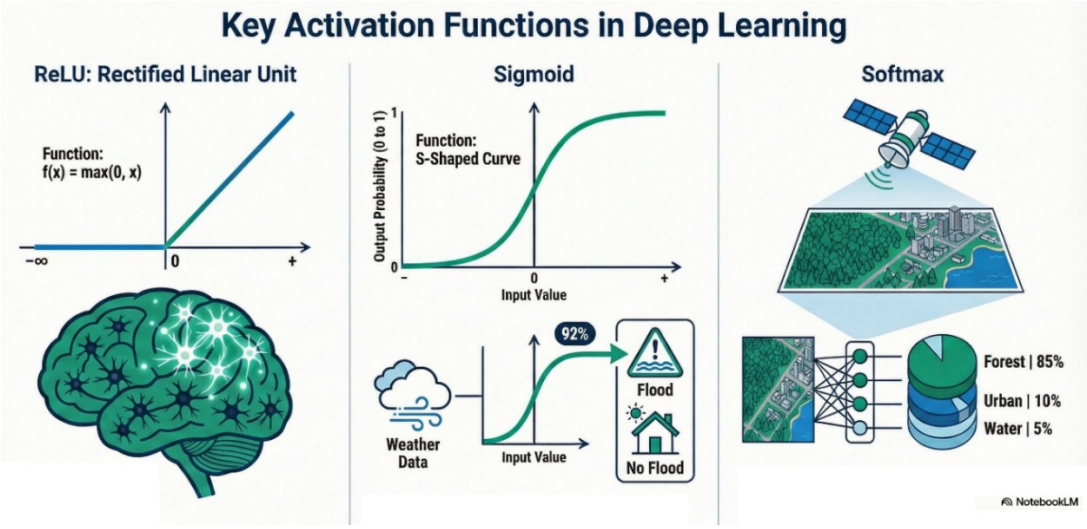
- **ReLU (**usmerjena linearna enota): opredeljena kot $f(x)=max(0,x)$, matematična funkcija, ki preprosto prepusti pozitivne signale, vse negativne pa popolnoma blokira.
  - Usmerjena linearna enota je trenutno »zlati standard« za skrite plasti pri globokem učenju.
  - Načelo delovanja: pozitivne signale prepusti nespremenjene, vse negativne signale pa popolnoma blokira.

- Računska učinkovitost: v nasprotju s kompleksnimi eksponentnimi funkcijami zahteva ReLU le preprosto preverjanje praga, zato jo je med učenjem mogoče izračunati izjemno hitro.
- Blaženje izginjajočih gradientov: v zelo globokih mrežah postajajo gradienti (signali, s katerimi se mreža »uči«) vedno manjši, dokler se mreža ne preneha učiti. ReLU[1] pomaga to preprečevati z ohranjanjem konstantnega gradienta za vse pozitivne vhode.
- Redkost: ker ReLU za vse negativne vhode poda vrednost nič, ustvarja »redke« predstavitve, pri katerih je v danem trenutku aktivna samo podmnožica nevronov. S tem posnema biološko učinkovitost človeških možganov.

• Sigmoidna funkcija

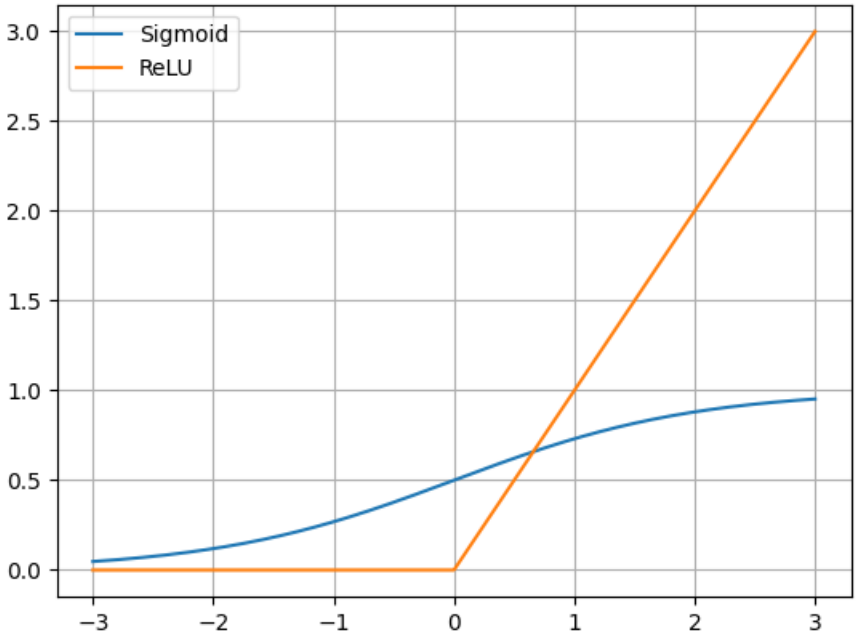
- Binarno razvrščanje**:** ker njen izhod ostaja v intervalu (0,1), je idealna izbira za **izhodno plast** modela, ki napoveduje binarni izid. *Primer:* napovedovanje verjetnosti »poplave« (>0.5) v primerjavi z »ni poplave« (<0.5).
- Zgodovinski kontekst: nekoč se je uporabljala v skritih plasteh, vendar jo je tam večinoma nadomestila ReLU, ker se sigmoidna funkcija »nasiči« (konca krivulje postaneta zelo ravna), kar lahko zaustavi učenje.

- **Softmax:**
  - Medtem ko sigmoidna funkcija obravnava »A v primerjavi z B«, je Softmax namenjena medsebojno izključujočim se kategorijam.
  - Verjetnostna porazdelitev: Softmax vzame vektor neobdelanih ocen iz končne plasti in jih stisne v verjetnostno porazdelitev. Ključna značilnost je, da je vsota vseh izhodov natanko 1.0 (100 %).
  - Okoljska uporaba: če iz satelitskih posnetkov razvrščamo pokrovnost tal, Softmax poskrbi, da model ne reče le, da je slikovna pika »večinoma gozd«, temveč poda natančno razčlenitev: 85 % gozd, 10 % urbano območje, 5 % voda.
  - Odločanje: za končno napoved modela se običajno izbere razred z največjo verjetnostjo.



Slika 6.4 Ključne aktivacijske funkcije

Prevod napisov na sliki: Ključne aktivacijske funkcije v globokem učenju: ReLU (usmerjena linearna enota), funkcija f(x) = max(0, x); sigmoidna funkcija, funkcija v obliki črke S, izhodna verjetnost (od 0 do 1), vhodna vrednost; vremenski podatki; poplava/brez poplave; Softmax: gozd 85 %, urbano 10 %, voda 5 %.



Slika 6.5 Primer sigmoidne aktivacijske funkcije in funkcije ReLU

Slika 6.5 prikazuje grafa sigmoidne aktivacijske funkcije in funkcije ReLU.

Učenje in optimizacija

Učenje nevronske mreže ni enkraten dogodek, temveč neprekinjen cikel poskusov in napak. Da bi mreža obvladala kompleksne okoljske podatke, sledi tristopenjskemu postopku: **širjenje naprej** (ugibanje), **izračun izgube** (merjenje napake) in **vzvratno razširjanje napake** (popravljanje notranjih uteži).

- **Funkcija izgube (ciljna funkcija):**

  Preden se lahko mreža izboljša, potrebuje matematično opredelitev tega, kaj je »napačno«. To je **funkcija izgube** (imenovana tudi ciljna funkcija). Deluje kot ena sama skalarna vrednost, ki predstavlja razdaljo med napovedjo mreže in dejanskimi referenčnimi podatki.

  - **Povprečna kvadratna napaka (MSE):** primerna za regresijske naloge, kot sta napovedovanje natančne koncentracije CO2 v ozračju ali temperature naslednjega dne. Razlike kvadrira, zato velike napake kaznuje bolj kot majhne.
  - **Izguba navzkrižne entropije:** običajno se uporablja za naloge razvrščanja (npr. »Ali ta slikovna pika satelitskega posnetka predstavlja gozd ali urbano območje?«). Meri, kako dobro se napovedana verjetnostna porazdelitev ujema z dejansko oznako.

- **Vzvratno razširjanje napake (konceptualni pregled):**

  Ko je izguba izračunana, mora mreža ugotoviti, katere konkretne uteži in odmiki so najbolj prispevali k napaki. To doseže z **vzvratnim razširjanjem napake**.

  Informacije tečejo nazaj od izhodne plasti proti vhodu[2]. Z uporabo matematičnih načel algoritem natančno določi, koliko je vsaka posamezna utež prispevala h končni napaki.

  - **Gradient** je v bistvu vektor, ki kaže v smeri največjega povečanja napake.
  - Z izračunom teh gradientov se mreža natančno nauči, kako bi rahlo povečanje ali zmanjšanje posamezne uteži vplivalo na končno napoved.

- **Gradientni spust:**

  Ko ima mreža gradiente, z optimizatorjem posodobi svoje uteži. Najpogostejša prispodoba za to je gradientni spust.

Predstavljajte si, da stojite na megleni gori (»pokrajini izgube«), vaš cilj pa je doseči najnižjo dolino (točko najmanjše napake). Doline ne vidite, lahko pa čutite naklon pod nogami.

- **Gradient:** pove vam, v kateri smeri je »navkreber«.
- **Korak:** da zmanjšate napako, naredite korak v *nasprotni* smeri gradienta (navzdol).
- **Hitrost učenja:** določa velikost vašega koraka. Če je korak prevelik, lahko preskočite dolino; če je premajhen, bo pot trajala večno.

- **Pogosti optimizatorji**

  Osnovni gradientni spust je temelj, sodobno globoko učenje pa za iskanje dna doline uporablja naprednejše »pohodnike«:

  - **Stohastični gradientni spust (SGD):** namesto da bi naenkrat upošteval celotno podatkovno zbirko, izračuna gradiente z majhnimi »serijami« podatkov, zato je postopek hitrejši in bolj dinamičen.
  - **Adam (Adaptive Moment Estimation):** trenutno najbolj priljubljen optimizator. Samodejno prilagaja hitrost učenja za vsako utež; upočasni, ko se približa rešitvi, in pospeši, ko je pot jasna.

Prekomerno prileganje in posploševanje pri okoljskih podatkih

Ključni izziv pri učenju nevronskih mrež je **prekomerno prileganje**. Do njega pride, ko se model učnim podatkom prilagodi *preveč* dobro in se namesto temeljnih fizikalnih zakonov nauči tudi naključnega šuma ali anomalij, značilnih za to podatkovno zbirko. **Heaton** poudarja, da bo imel prekomerno prilagojen model majhno napako pri učnih podatkih, vendar veliko napako pri novih, še nevidenih podatkih (slabo **posploševanje**).

Okoljska posledica:
Prekomerno prilagojen model, naučen na vremenskih podatkih iz obdobja 2010–2020, bi si lahko zapomnil posamezne nevihte iz teh let. Če bi ga uporabili leta 2025, morda ne bi napovedal suše, ker bi iskal natančne vzorce iz leta 2010 namesto splošnih podnebnih kazalnikov.

Strategije blaženja:

- **Izpuščanje (dropout):** tehnika, pri kateri med učenjem naključno izbrane nevrone prezremo (izpustimo). Tako je mreža prisiljena naučiti se robustnih značilk, namesto da bi se zanašala na posamezne poti.
- **Zgodnje ustavljanje:** spremljanje delovanja modela na ločeni »validacijski množici« in takojšnja ustavitev učenja, ko se začne validacijska napaka povečevati, tudi če se učna napaka še vedno zmanjšuje.

Modeliranje zaporedij: od MLP do LSTM

V nasprotju s tradicionalnimi algoritmi ali osnovnimi mrežami s širjenjem naprej, ki podatkovne točke večinoma obravnavajo neodvisno, so okoljski podatki sami po sebi zaporedni. Za učinkovito obdelavo takih podatkov uporabljamo mrežo z **dolgim kratkoročnim spominom (LSTM)** [2].

LSTM je specializirana rekurentna nevronska mreža (RNN), zasnovana tako, da si »zapomni« dolgoročne trende in reši matematični izziv ohranjanja informacij v daljših obdobjih (t. i. problem »izginjajočega gradienta«). Predstavljajte si tekoči trak, ki poteka vzporedno z zaporedjem, ki ga obdelujete. Informacije iz preteklosti lahko skočijo na ta tekoči trak, se prenesejo čez številne časovne korake in po potrebi z njega izstopijo nespremenjene. LSTM uporablja sistem učljivih **vrat** (vhodnih, izhodnih in pozabnih), ki uravnavajo ta tok ter natančno odločajo, katere informacije naj se ohranijo in katere zavržejo.

Ta arhitektura je idealna za okoljske aplikacije in omogoča globljo analizo kot preprosta regresija. Namesto da bi model LSTM za napoved današnjega dne uporabil samo včerajšnje podatke, pregleda **zaporedje zadnjih 14 dni** in pred napovedjo prepozna trende segrevanja ali ohlajanja.

Osrednje arhitekture globokega učenja v okoljskih aplikacijah

Ta razdelek obravnava temeljne arhitekture nevronskih mrež, ki se uporabljajo za modeliranje okoljskih sistemov. Namesto na kompleksne matematične izpeljave se osredotočamo na **intuitivno razumevanje** teh modelov: *kdaj* uporabiti posamezno arhitekturo glede na strukturo okoljskih podatkov (npr. tabelarični odčitki senzorjev, satelitski posnetki ali zgodovinske vremenske časovne vrste).

Mreže s širjenjem naprej (FFN): temelj

Mreža s širjenjem naprej, pogosto imenovana večplastni perceptron (MLP), je značilni model globokega učenja. V teh mrežah informacije tečejo v eno smer – od vhodnih vozlišč prek skritih plasti do izhodnih vozlišč – brez zank.

Kako deluje: FFN približujejo funkcije tako, da vhodne vektorje preslikajo v ciljne izhode. Zelo uspešne so pri učenju globalnih vzorcev v tabelaričnih podatkih.

Okoljska uporaba: FFN so primerne za **statične napovedne naloge**, pri katerih vrstni red podatkovnih točk ni pomemben ali pa so časovne odvisnosti šibke.

Primer: Napovedovanje trenutne ravni onesnaženosti na določeni lokaciji na podlagi trenutnih vrednosti hitrosti vetra, temperature in gostote prometa.

Omejitev: FFN nimajo "spomina". Vsak vhod obdelajo neodvisno, zato so manj učinkovite pri podatkih, pri katerih je pomembno zaporedje, kot so dolgotrajni vročinski valovi ali sezonski trendi.

Konvolucijske nevronske mreže: obdelava prostorskih podatkov

Medtem ko FFN vhodne značilke obravnavajo neodvisno, imajo okoljski podatki pogosto mrežno topologijo, kot jo imajo satelitski posnetki ali radarski zemljevidi. CNN so specializirane za obdelavo takšnih podatkov.

Kako deluje: Namesto povezovanja vsakega nevrona z vsakim drugim nevronom CNN uporabljajo majhne "filtre" (jedra), ki drsijo čez vhodno sliko in zaznavajo lokalne vzorce, kot so robovi, teksture ali oblike. Tako lahko model prepozna značilko (npr. drevo ali stavbo) ne glede na to, kje se pojavi na sliki (invariantnost glede na premik).

Okoljska uporaba: CNN so standard pri **daljinskem zaznavanju**. *Primer:* analiziranje satelitskih posnetkov za kartiranje krčenja gozdov v Amazoniji ali razvrščanje zdravja koralnih grebenov na podlagi podvodnih fotografij.

Rekurentne arhitekture (RNN, LSTM, GRU): analiza časovnih vrst

Okoljski podatki so po naravi zaporedni; poznavanje včerajšnje temperature je ključno za napovedovanje današnje temperature. Rekurentne nevronske mreže (RNN) to upoštevajo z uvedbo notranje zanke, ki omogoča ohranjanje informacij.

Standardne RNN: Te mreže ohranjajo "stanje", ki vsebuje informacije o tem, kar so videle do danega trenutka. Vendar standardne RNN trpijo zaradi "problema izginjajočega gradienta", zato

si ne morejo zapomniti informacij v dolgih zaporedjih (npr. povezati padavinskega dogodka izpred dveh tednov z današnjim vodostajem reke).

Mreža z dolgim kratkoročnim spominom (LSTM): Za rešitev težave s spominom so bile uvedene mreže LSTM s specializirano notranjo strukturo, ki vključuje "vrata" (vhodna vrata, vrata pozabljanja in izhodna vrata). Ta vrata mreži omogočajo, da se izrecno nauči, kaj naj si v dolgih obdobjih zapomni in kaj naj pozabi.

Rekurentna enota z vrati (GRU): Poenostavljena različica LSTM, ki jo je pogosto mogoče hitreje učiti, pri številnih nalogah z okoljskimi časovnimi vrstami pa dosega primerljivo uspešnost. Okoljska uporaba: te arhitekture so ključne za napovedovanje.

Primer: Napovedovanje poplavne ogroženosti na podlagi zaporedja podatkov o padavinah in vlažnosti tal v zadnjih 14 dneh ali modeliranje podnebnih trendov, pri katerih zgodovinski kontekst določa prihodnja stanja.

Odgovorna umetna inteligenca: prekomerno prileganje in etika

Uvajanje globokega učenja v resnično okolje zahteva več kot le visoko točnost; zahteva zanesljivost in etično odgovornost.

Preprečevanje prekomernega prileganja: Pogosta past je izdelava modela, ki si zapomni učne podatke (npr. določene vremenske anomalije iz leta 2010), vendar ne zmore posploševati na prihodnje podatke. Da bi zagotovili robustnost modelov, uporabljamo tehnike, kot sta izpuščanje (**Dropout**; naključno izpuščanje nevronov med učenjem, da se prepreči odvisnost od določenih poti) in zgodnje ustavljanje (**Early Stopping**; ustavitev učenja, ko se uspešnost na validacijskih podatkih poslabša).

Etična in razložljiva umetna inteligenca: Modeli globokega učenja so pogosto kritizirani kot "črne skrinjice". V okoljski politiki si moramo prizadevati za preglednost. Tehnologija ni nikoli nevtralna; tehnične odločitve so tudi etične odločitve. Kadar modeli vplivajo na odločitve – na primer na razporejanje vode ali odzivanje ob nesrečah – je ključno preveriti, da ne ohranjajo pristranskosti iz zgodovinskih podatkov in da lahko njihove napovedi razložijo področni strokovnjaki.

Od globokega učenja do odločanja, prilagojenega podnebnim spremembam

Ko je model globokega učenja naučen in validiran, se pozornost s tehnične optimizacije preusmeri na **operativni učinek**. Doseganje nizke stopnje napake (na primer z uporabo metrik, kot sta povprečna absolutna napaka ali koren povprečne kvadratne napake) je tehnični mejnik, vendar je končni cilj pretvoriti te številčne napovedi v smiselne ukrepe za prilagajanje podnebnim spremembam. Napovedno modeliranje podpira tri ključne stebre trajnosti:

- **Proaktivno načrtovanje.** Uporaba dolgoročnih napovedi za načrtovanje odporne infrastrukture (npr. dimenzioniranje odvodnih sistemov na podlagi napovedane intenzivnosti padavin).
- **Zmanjševanje tveganja:** Uvedba sistemov zgodnjega opozarjanja, pri katerih izhodi modela neposredno sprožijo varnostne protokole.
- **Razporejanje virov:** Optimiziranje porazdelitve vode ali energije na podlagi pričakovanih okoljskih razmer.

Napovedno modeliranje za prilagajanje podnebnim spremembam

Globoko učenje nadomešča počasne fizikalne simulacije s hitrimi podatkovno vodenimi modeli in tako omogoča prilagajanje podnebnim spremembam v realnem času na ključnih področjih:

- Napovedovanje ekstremnih vremenskih vzorcev (vročinskih valov, neviht, suš)
- Napovedovanje hidroloških sprememb (poplav, spremenljivosti rečnega pretoka, pomanjkanja vode)
- Modeliranje verjetnosti in širjenja požarov v naravi
- Predvidevanje poslabšanja kakovosti zraka in vplivov na zdravje ljudi.

Ti modeli omogočajo podnebno pametno načrtovanje, kot so razvoj odporne infrastrukture, sistemi zgodnjega opozarjanja in pripravljenost na odziv ob nesrečah. Napovedi globokega učenja omogočajo natančno prilagajanje podnebnim spremembam, saj izhode modelov povezujejo z določenimi sprožilci ukrepov in ukrepi za krepitev odpornosti. Spodaj so razširjeni primeri, strukturirani kot **(Zaznava → Ukrep)**, s podrobnostmi o delovanju modelov, uporabi v resničnem svetu in rezultatih prilagajanja, povzetimi iz uveljavljenih študij primerov.

Primeri scenarijev uporabe za prilagajanje podnebnim spremembam:

- **Uporabe za napovedovanje poplav**
  - **(Zaznava)** Modeli CNN-LSTM obdelujejo satelitske posnetke in podatke merilnikov plimovanja, da napovejo verjetnost obalnih poplav v naslednjih 48 urah, pri scenarijih dviga morske gladine pa dosegajo 20 % večjo točnost kot tradicionalni modeli. → **(Ukrep)** Občine aktivirajo prilagodljivo zaščito obale, kot je obnova mangrov, in dvignejo kritično infrastrukturo; zgodnja opozorila zmanjšajo gospodarsko škodo za 30–50 % z vnaprejšnjo razporeditvijo vreč s peskom in preusmerjanjem prometa [5],
  - **(Zaznava)** Prostorsko-časovne mreže LSTM združujejo radarske podatke o padavinah in podatke rečnih merilnikov za kartiranje globine mestnih poplav v realnem času z urno ločljivostjo. → **(Ukrep)** Reševalne službe prebivalcem ponudijo aplikacije z dinamičnimi opozorili, optimizirajo delovanje črpališč in posodobijo mestne prostorske predpise s standardi za poplavno odporno gradnjo [6].
- **Obvladovanje tveganja požarov v naravi**
  - **(Zaznava)** Ansambelski modeli CatBoost-RNN analizirajo vegetacijske indekse, vzorce vetra in temperaturo iz satelitskih podatkov MODIS, da napovejo dnevno verjetnost nastanka požara v naravi na območjih, velikih 10.000 km². → **(Ukrep)** Gozdarske službe načrtujejo nadzorovane požige v obdobjih majhnega tveganja, pošiljajo stopnjevana navodila za evakuacijo prek SMS in preventivno razporejajo letala za gašenje z vodo, s čimer stroške gašenja zmanjšajo za 25 %. [7],
  - **(Zaznava)** Hibridni okviri CNN-BiLSTM simulirajo hitrost širjenja požara z uporabo podatkov VIIRS o zaznanih aktivnih požarih in topografskih podatkov za skoraj sprotno napovedovanje obsega požara. → **(Ukrep)** Skupnosti namestijo omrežja senzorjev za zgodnje zaznavanje dima, razširijo protipožarne preseke z avtohtonim rastlinjem in usposobijo prostovoljne enote za hitro posredovanje [8].
- **Strategije za suše in vročinske valove**
  - **(Zaznava)** Hibridni modeli globokega učenja QRNN+wavelet napovedujejo indeks suše pretoka (SDI-3, SDI-6, SDI-12) za Stockholm na Švedskem ter vključujejo razčlenitev časovnih vrst za natančnost na več časovnih skalah. → **(Ukrep)** Vodni organi uvedejo omejitve 3–12 mesecev vnaprej, spremenijo dolgoročne pogodbe o dobavi in prilagodijo delovanje hidroelektrarn, da ublažijo napovedani primanjkljaj pretoka [9],

- **(Zaznava)** Globoko in ansambelsko učenje na podlagi evropskih podnebnih simulacij prepoznava epizode ekstremne vročine in suše, ki krepijo vročinske valove v južni in zahodni Evropi, ter izpostavlja glavne podnebne kazalnike. → **(Ukrep)** Mesta, kot sta Lizbona in Atene, aktivirajo "načrte za izredne razmere ob vročini": hladilne centre, prepoved dela na prostem in ciljno usmerjena opozorila SMS za starejše prebivalce [10].

Študija primera: napovedovanje vročinskih valov z globokimi nevronskimi mrežami

Vir podatkov Open-Meteo

V tej študiji primera je **platforma Open-Meteo** uporabljena kot referenčni vir vremenskih podatkov. Open-Meteo prek javno dostopnega API-ja zagotavlja brezplačen dostop do zgodovinskih vremenskih podatkov za katero koli lokacijo po svetu ter uporabnikom omogoča izbiro časovnega obdobja in nabora vremenskih spremenljivk.

Za učenje modela globokega učenja, ki lahko napoveduje vročinske valove, potrebujemo določene spremenljivke, ki vplivajo na toplotno obremenitev. Prenos je treba opraviti prek omrežnega vmesnika ter določiti geografsko lokacijo (zemljepisno širino in dolžino), časovno ločljivost (dnevno) in izbrane meteorološke spremenljivke.

V tej vaji se bomo osredotočili na ključno spremenljivko temperature_2m_max – to je ciljna spremenljivka, ki jo želimo napovedati za prepoznavanje dogodkov ekstremne vročine.

Podatki se neposredno uvozijo v okolje Python z uporabo knjižnice Pandas, ki odziv API-ja v obliki **CSV** razčleni v strukturiran nabor podatkov, primeren za strojno učenje.

Primer globokega učenja

V tej praktični vaji prehajamo od preproste statistike k sodobnemu globokemu učenju. Zgradili bomo sistem, ki ne bo le povedal "danes je vroče", temveč bo z analiziranjem podnebne dinamike skozi čas dejavno opozarjal na bližajoče se nevarne vročinske valove.

Za to uporabimo mrežo **LSTM (dolgi kratkoročni spomin)** – nekakšne umetne možgane, ki lahko v času odkrivajo vzročno-posledične povezave.

Vodnik po korakih: kako zagnati simulacijo

Uporabili bomo **Google Colab**, brezplačno platformo v oblaku, ki omogoča izvajanje tega modela umetne inteligence neposredno v brskalniku brez nameščanja programske opreme.

1. **Odprite Google Colab:**
   - V spletnem brskalniku odprite colab.research.google.com,
   - Prikazalo se bo okno "Welcome to Colaboratory".
2. **Naložite zvezek:**
   - V pojavnem oknu kliknite zavihek **Upload**,
   - Datoteko "*ML_DL.ipynb*" povlecite in spustite v polje (ali kliknite "Choose File" in jo izberite v računalniku).
3. **Zaženite kodo:**
   - Ko se zvezek naloži, poglejte zgornjo menijsko vrstico,
   - Kliknite **Runtime** in izberite **Run all** (ali pritisnite Ctrl+F9),
   - *Opomba:* Morda se prikaže opozorilo "Notebook requires high RAM" ali "Author not trusted" – preprosto kliknite **Run anyway**.
4. **Uporabite interaktivne elemente:**
   - Med izvajanjem kode se pomikajte navzdol. Videli boste nalaganje podatkovnih tabel, premikanje kazalnikov napredka učenja in na koncu prikaz interaktivnih grafov.

Tehnični predpogoji

Izvedba poteka v jeziku Python z uporabo naslednjih specializiranih knjižnic Python:

- **TensorFlow/Keras**: za gradnjo in učenje nevronske mreže **LSTM** (dolgi kratkoročni spomin),
- **Scikit-learn**: za normalizacijo podatkov (skaliranje) in izračun metrik uspešnosti (MAE, RMSE),
- **Pandas/NumPy**: za obdelavo podatkov časovnih vrst in izvajanje številčnih izračunov,
- **Folium/Plotly/Seaborn**: za ustvarjanje interaktivnih grafov napovedi in geoprostorskih zemljevidov za podporo odločanju.

Vir podatkov

Platformo **Open-Meteo** uporabljamo kot svoj "časovni stroj". Zagotavlja zgodovinske vremenske podatke za katero koli lokacijo na Zemlji. Za učenje umetne inteligence smo izbrali ključno spremenljivko “Najvišja temperatura (2 m)” – ciljno spremenljivko, ki jo želimo napovedati.

Potek zvezka: razumevanje programske kode

Priloženi zvezek “ML_DL.ipynb” je celovit, popolnoma delujoč delovni potek, pripravljen za zagon. Vnaprej je nastavljen tako, da samodejno prenese podatke, nauči nevronsko mrežo in ustvari interaktivna poročila. Za ogled rezultatov ni potrebno dodatno programiranje.

Za prilagoditev simulacije različnim mestom ali scenarijem morajo študenti spremeniti le **blok 1 (Konfiguracija in knjižnice).** Blokov od 2 do 8 **ni treba spreminjati**; vsebujejo osrednjo logiko (obdelavo podatkov, učenje LSTM, vizualizacijo). Ti bloki so avtomatizirani in jih je treba le zaporedno izvesti.

Praktično delo je razdeljeno na 8 logičnih blokov:

1. Konfiguracija in knjižnice. V tem prvem koraku nastavimo "nadzorno ploščo" poskusa. Pred zagonom umetne inteligence lahko študenti spremenijo določene spremenljivke in simulacijo prilagodijo različnim mestom ali scenarijem:

- **Geografska lokacija:** LATITUDE & LONGITUDE: Koordinate ciljnega mesta (privzeto je to Kijev v Ukrajini: 50.45, 30.52). Z njuno spremembo lahko preizkusite delovanje modela v povsem drugačnih podnebjih, na primer v deževnem Šibeniku (Hrvaška), suhem Rimu (Italija) ali Skopju (Severna Makedonija).
- **Nastavitve časovnega stroja (obseg podatkov)**: START_DATE & END_DATE. Zgodovinsko obdobje, uporabljeno za učenje umetne inteligence (npr. od "2010-01-01" do "2025-12-31"). Več let zgodovinskih podatkov modelu praviloma pomaga pri učenju boljših vzorcev.
- **Obseg "spomina" umetne inteligence:** LOOKBACK_DAYS. Velikost "drsečega okna" (privzeto **14 dni**). Določa, kako daleč v preteklost model pogleda pri oblikovanju napovedi.
- **Nadzor vizualizacije:** DISPLAY_START_DATE & DISPLAY_END_DATE. Ti spremenljivki določata raven povečave glavnega interaktivnega grafa (6. korak). Namesto prikaza vseh 15 let se lahko osredotočimo na določeno zanimivo leto (npr. 2025).
- **Varnostni pragovi (sistem odločanja):**

- HEATWAVE_WARNING: Temperaturni prag (npr. **27.0 °C**), ki sproži "oranžno" opozorilo za javnost.
- **o** CRITICAL_EMERGENCY: Ekstremna temperatura (npr. 31.0 °C), ki sproži "rdeče" kritično opozorilo za reševalne službe.

- **Obdobje analize napovedi:** DISPLAY_FORECAST_START_DATE & DISPLAY_FORECAST_END_DATE. Določen obseg (npr. avgust 2025), uporabljen v 7. koraku za preverjanje, ali je opozorilni sistem v določenem mesecu pravilno prepoznal vročinske valove.

2. Nalaganje podatkov. Koda se samodejno poveže z API-jem Open-Meteo in prenese vremensko zgodovino za zadnjih 15 let. Študenti lahko opazujejo, kako se "surovi" podatki pretvorijo v strukturirano tabelo Pandas (slika 6.6).

```
Fetching data from Open-Meteo...
Data loaded: 5844 records.
```

	Date	MaxTemp	SolarRad
0	2010-01-01	-6.2	1.83
1	2010-01-02	-6.2	2.66
2	2010-01-03	-11.4	3.29
3	2010-01-04	-12.4	3.87
4	2010-01-05	-8.7	3.60

Slika 6.6 Podatki, pridobljeni s platforme Open-Meteo

Prevod napisov na sliki: Pridobivanje podatkov iz Open-Meteo; naloženih 5844 zapisov. Stolpci: datum, najvišja temperatura in sončno sevanje.

3. Predobdelava: "drseče okno". To je ključna razlika v primerjavi s klasičnimi algoritmi. Namesto učenja modela po načelu *"Včeraj je bilo toplo → danes bo toplo"* ustvarimo **14-dnevno** drseče okno (obdobje pogleda nazaj). Model opazuje zaporedje (1. dan ... 14. dan), da napove 15. dan. *Zakaj:* Tako *lahko* nevronska mreža zazna *vztrajnost* podnebja – kako se vročina pred vrhuncem postopoma kopiči.

4. Gradnja in učenje modela LSTM (slika 6.7**)**. Zgradimo "možgane" svojega sistema:

- Uporabimo **plast LSTM**, ki deluje kot spomin sistema, ohranja pomembne trende in zavrača naključni šum.
- Dodan je mehanizem izpuščanja (**Dropout**) za preprečevanje **prekomernega prileganja** – pojava, pri katerem si model preprosto zapomni pretekla leta, vendar ne razume splošnih zakonitosti vremena.

```
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - loss: 0.0029 - mae: 0.0414 - val_loss: 0.0025 - val_mae: 0.0388
Epoch 38/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.0030 - mae: 0.0423 - val_loss: 0.0025 - val_mae: 0.0388
Epoch 39/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - loss: 0.0029 - mae: 0.0413 - val_loss: 0.0025 - val_mae: 0.0389
Epoch 40/50
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - loss: 0.0030 - mae: 0.0423 - val_loss: 0.0025 - val_mae: 0.0389
Epoch 40: early stopping
Restoring model weights from the end of the best epoch: 35.
Training complete.
```

Slika 6.7 Učenje LSTM z zgodnjim ustavljanjem (Early Stopping)

Prevod napisov na sliki: Epoha 40: zgodnje ustavljanje; obnovitev uteži modela s konca najboljše epohe 35; učenje končano.

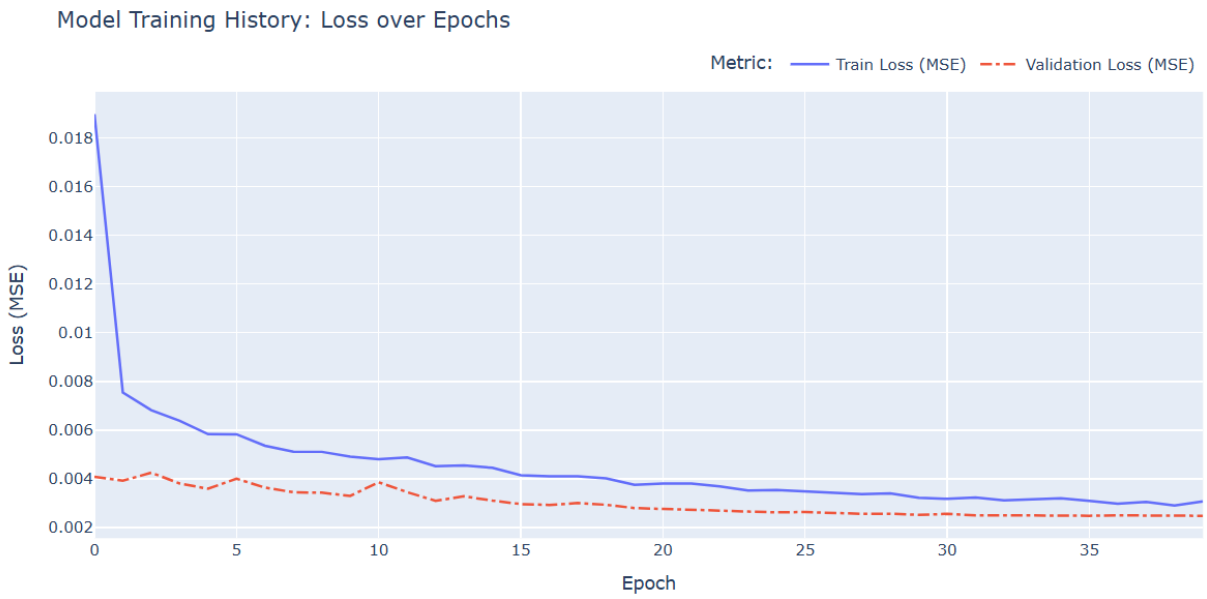
5. Vizualizacija uspešnosti učenja. Prikažemo "krivuljo učenja".

Cilj: želimo, da se obe črti sčasoma pomikata navzdol. To pomeni, da se napaka modela (Loss) zmanjšuje.

- **Modra** črta **(Train Loss):** kako dobro se model uči iz "učnega gradiva" (zgodovinskih podatkov),
- **Rdeča črta (**Validation **Loss):** kako uspešen je model pri "novih vprašanjih", ki jih še ni videl.

Kaj opazovati na sliki 6.8:

- **Dober rezultat:** obe črti se pomikata navzdol in ostajata blizu druga drugi.
- **Prekomerno prileganje:** če se modra črta pomika navzdol, rdeča pa se začne pomikati **navzgor**, si model podatke zapomni, namesto da bi se učil vzorcev.



Slika 6.8 Zgodovina učenja modela: izguba po epohah

Prevod napisov na sliki: Zgodovina učenja modela: izguba po epohah; učna izguba (MSE); validacijska izguba (MSE).

6. Vrednotenje in interaktivna napoved. Za ustvarjanje interaktivnega grafa uporabimo knjižnico **Plotly**.

6a. Poročilo o uspešnosti modela (metrike):

Pred ogledom grafov izračunamo matematično oceno modela (slika 6.9).

- **MAE (**povprečna **absolutna napaka):** Povprečno "odstopanje" v stopinjah (npr. če se model običajno zmoti za 1,5 °C).
- **RMSE (koren povprečne kvadratne napake):** Metrika, ki velike napake kaznuje bolj kot majhne.
- **R^2 (ocena točnosti):** Odstotkovna ocena (od 0.0 do 1.0), ki kaže, kolikšen delež temperaturne spremenljivosti model uspešno pojasni.

```
--- 📊 WEATHER MODEL REPORT CARD ---

1. Average Error (MAE): 2.14°C
   👉 Explanation: On average, the model's prediction is off by about 2.1°C.
--------------------------------------------------
2. Penalty for Big Mistakes (RMSE): 2.74°C
   👉 Explanation: This metric punishes large errors more heavily.
      If this number is much higher than the Average Error, the model occasionally makes big misses.
--------------------------------------------------
3. Accuracy Score (R²): 0.93 (or 93.1%)
   👉 Explanation: How much of the temperature variation does the model explain?
      (1.0 = Perfect, 0.0 = Useless, Negative = Worse than guessing)
   ✅ Quality: High Confidence.
--------------------------------------------------
```

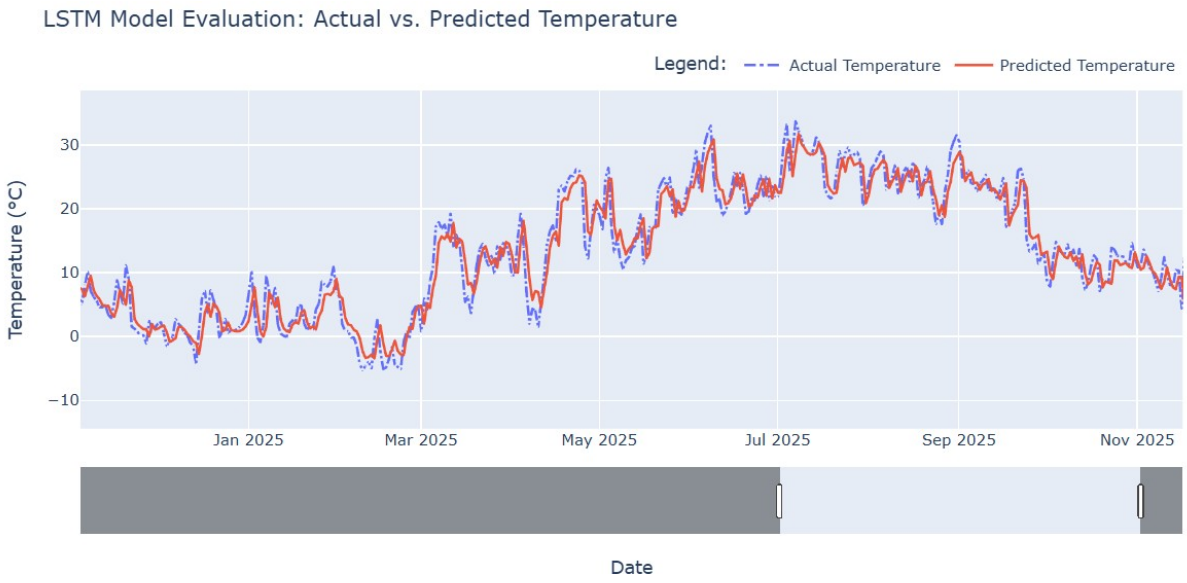
Slika 6.9 Poročilo o uspešnosti vremenskega modela

Prevod napisov na sliki: Poročilo vremenskega modela: 1. povprečna napaka (MAE) 2,14 °C – napoved modela v povprečju odstopa za približno 2,1 °C; 2. kazen za velike napake (RMSE) 2,74 °C – velike napake kaznuje izraziteje; če je ta vrednost precej višja od povprečne napake, model občasno naredi velike napake; 3. ocena točnosti (R^2) 0,93 oziroma 93,1 % – kolikšen delež temperaturne spremenljivosti pojasni model (1,0 = popolno, 0,0 = neuporabno, negativno = slabše od ugibanja); kakovost: visoka stopnja zaupanja.

6b. Napovedi (interaktivna napoved)

S tem s knjižnico Plotly ustvarimo dinamičen **graf časovne vrste** (slika 6.10).

- **Rdeča** črta**:** temperatura, ki jo napove umetna inteligenca,
- **Modra** črtkana **črta:** dejansko izmerjena temperatura,
- Interaktivni **drsnik:** na dnu študentom omogoča približanje prikaza posameznih tednov ali mesecev, da preverijo, ali rdeča črta natančno sledi nenadnim vrhovom vročinskega vala.



Slika 6.10 Vrednotenje modela LSTM: dejanska in napovedana temperatura

Prevod napisov na sliki: Vrednotenje modela LSTM: dejanska in napovedana temperatura; legenda: dejanska temperatura (modra črtkana črta), napovedana temperatura (rdeča črta); temperatura (°C), datum.

6c. Dejanske in napovedane vrednosti (dobrota prileganja)

S tem ustvarimo **raztreseni diagram** za preverjanje pristranskosti.

- Os **X:** dejanska temperatura.
- **Os Y:** napovedana temperatura.
- **»Črta popolnega ujemanja«:** diagonalna rdeča črtkana črta. Če bi bil model popoln, bi vsaka modra pika ležala natanko na tej rdeči črti. Ta prikaz nam pomaga ugotoviti, ali model temperaturo dosledno precenjuje (pike nad črto) ali podcenjuje (pike pod črto).

6d. Diagram ostankov

S tem ustvarimo diagram, ki prikazuje **ostanke** (oziroma napake) skozi čas.

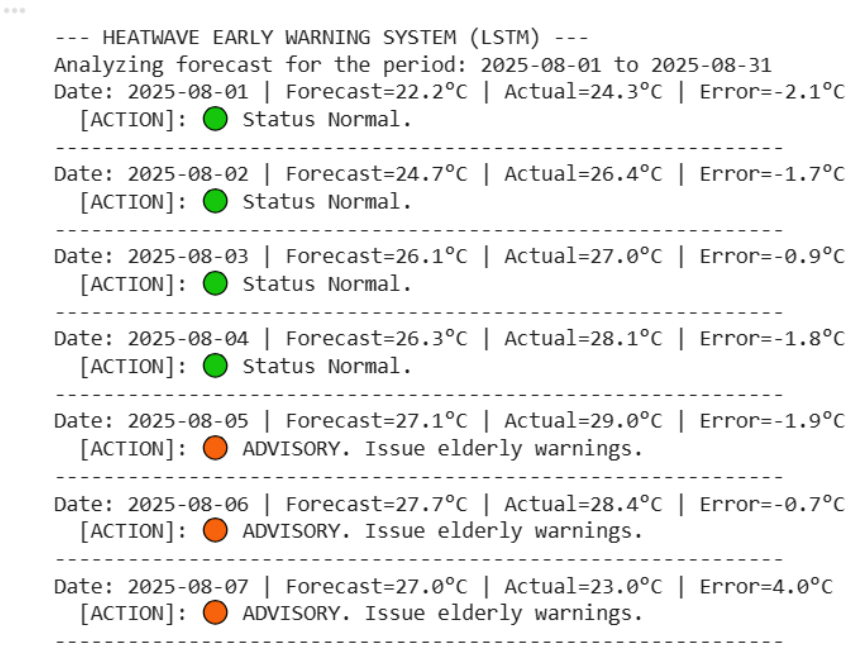
- **Ničelna črta:** predstavlja popolno napoved (napaka 0),
- **Namen:** iščemo naključnost. Če so pike videti kot naključni »šum« okoli ničelne črte, je model ustrezen. Če opazimo jasne ponavljajoče se valovne vzorce, model ne upošteva nekega ključnega dejavnika (na primer sezonskosti).

7. Sistem odločanja, prilagojen podnebnim spremembam

V tem bloku modeliramo delovanje **sistema zgodnjega opozarjanja**. Umetna inteligenca ne napoveduje le temperature, temveč deluje kot »digitalni svetovalec« mestni upravi in sporoča, kdaj je treba aktivirati varnostne protokole.

Sistem deluje po načelu semaforja in napoved UI primerja z varnostnimi pragovi (THRESHOLDS), ki smo jih določili v **bloku 1**.

- ● Stopnja »Normalno« (zelena)
  - Pogoj: napovedana temperatura je pod opozorilnim pragom (na primer < 30 °C).
  - Ukrepi skupnosti:
    - Redno spremljanje vremenskih razmer,
    - Običajno delovanje občinskih služb,
    - *Pojasnilo:* razmere so varne; ukrepanje ni potrebno.
- ● Stopnja »Obvestilo/opozorilo« (oranžna)
  - Pogoj: temperatura preseže prvi prag (na primer ≥ 30 °C, spremenljivka HEATWAVE_WARNING),
  - Ukrepi skupnosti:
    - Obveščanje: pošiljanje opozorilnih sporočil SMS ranljivim skupinam prebivalstva (starejšim, kroničnim bolnikom),
    - Zdravstvo: uvedba visoke stopnje pripravljenosti reševalnih služb,
    - Socialne službe: socialni delavci dodatno obiščejo starejše prebivalce, ki živijo sami.
- ● Stopnja »Kritično izredno stanje« (rdeča)
  - Pogoj: temperatura doseže skrajne vrednosti (na primer ≥ 35 °C, spremenljivka CRITICAL_EMERGENCY).
  - Ukrepi skupnosti:
    - Infrastruktura: odprtje občinskih »centrov za ohlajanje« – klimatiziranih prostorov v knjižnicah, šolah ali skupnostnih centrih, dostopnih javnosti,
    - Varnost pri delu: omejitev ali prepoved težkega dela na prostem (gradbena dela, popravila cest) v času največje pripeke,
    - Oskrba z vodo: postavitev mobilnih postaj za razdeljevanje pitne vode na mestnih trgih,
    - Električno omrežje: priprava na morebitne rotacijske izklope zaradi preobremenitve omrežja, ki jo povzročajo klimatske naprave.



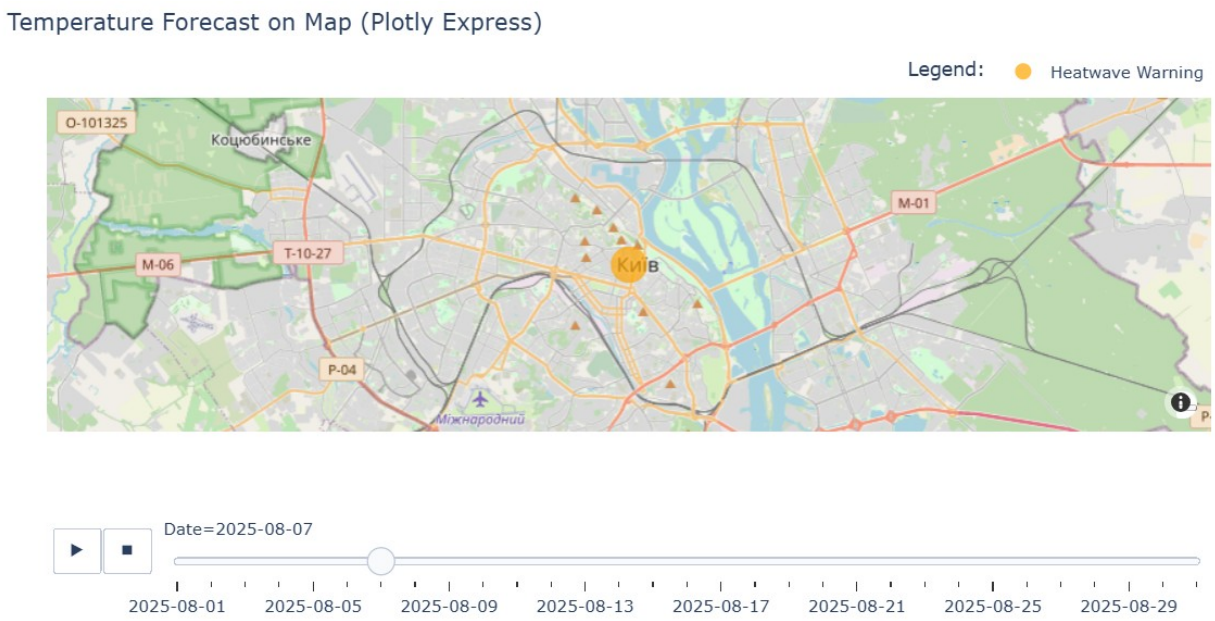
Slika 6.11 Sistem zgodnjega opozarjanja na vročinske valove

Prevod napisov na sliki: Sistem zgodnjega opozarjanja na vročinske valove (LSTM): datum, napoved, dejanska temperatura, napaka in ukrep; normalno stanje ali opozorilo starejšim.

8. Geoprostorski prikaz

V tem zadnjem bloku časovno napoved preoblikujemo v **geoprostorsko orodje za podporo odločanju**. Pike na zemljevidu spreminjajo barvo (modra/oranžna/rdeča) glede na napovedano stopnjo nevarnosti. Tako pokažemo, kako lahko mestni uradniki spremljajo posamezne mestne četrti. Statični zemljevidi običajno prikazujejo posnetek stanja, ta animirani zemljevid pa prikazuje *razvoj dogajanja*.

Uporaba v skupnosti: če bi ta model razširili na omrežje senzorjev (npr. mestno središče, predmestje, industrijska cona), bi lahko mestna uprava videla, katere četrti se pregrevajo (učinek urbanega toplotnega otoka), in cisterne z vodo poslala *samo* v te četrti, s čimer bi prihranila vire.



Slika 6.12 Prikaz na zemljevidu na podlagi parametrov CRITICAL_EMERGENCY in HEATWAVE_WARNING

Prevod napisov na sliki: Napoved temperature na zemljevidu (Plotly Express); legenda: opozorilo na vročinski val; datum.

Naloge vaje

V tej vaji boste z globokim učenjem napovedovali vzorce ekstremnega vremena v mestu po lastni izbiri. Nato boste rezultate uporabili za razpravo o praktičnih rešitvah za zaščito mestne infrastrukture in prebivalstva.

1. naloga: geografska prilagoditev

- **Odprite** priloženo datoteko zvezka v okolju Google Colab.
- V **bloku 1 poiščite** spremenljivki LATITUDE in LONGITUDE.
- Koordinate **spremenite** tako, da bodo ustrezale mestu po vaši izbiri (npr. vašemu domačemu kraju ali vročemu mestu, kot so Rim, Atene in druga).
- **Zaženite** zvezek, da ustvarite nove napovedi.
- **Analizirajte rezultate sistema odločanja v bloku 7:**
  - Ali sistem izpiše preveč sporočil **»CRITICAL ALERT«** (rdeče)? (To nakazuje, da so pragovi za to vroče podnebje prenizki, kar bi lahko povzročilo »utrujenost zaradi opozoril«.)

- Ali pa kljub vročemu vremenu sploh ni opozoril? (To nakazuje, da so pragovi morda previsoki.)

- **Prikaz na zemljevidu (blok 8):** pomaknite se navzdol do interaktivnega zemljevida. Ali vidite RDEČE ali ORANŽNE pike? Ali se prostorska razporeditev »nevarnih območij« ujema z vašimi pričakovanji za to mesto?
- **Umerjanje:** vrnite se v blok 1 in prilagodite spremenljivki HEATWAVE_WARNING in CRITICAL_EMERGENCY, da bosta bolje ustrezali podnebju izbranega mesta.
- **Znova zaženite zvezek.**

Vprašanje: kako je sprememba pragov izboljšala uporabnost sistema zgodnjega opozarjanja za to lokacijo? Katero **politiko urbanističnega načrtovanja** bi predlagali za zmanjšanje tveganja zaradi vročine na tem območju v naslednjih petih letih? (Npr. sajenje več dreves, nameščanje zelenih streh.)

2. naloga: preizkus »spomina« (uglaševanje hiperparametrov)

- V **bloku 1 poiščite** spremenljivko LOOKBACK_DAYS = 14. Ta predstavlja »obseg spomina« UI (koliko preteklih dni analizira).
- Vrednost **spremenite** s 14 na 3 dni.
- **Znova zaženite** celoten zvezek (Runtime -> Run all), da model ponovno naučite s tem »kratkoročnim spominom«.
- **Pomaknite se navzdol do grafa vrednotenja v bloku 6b.**

Vprašanje: primerjajte obnašanje rdeče črte (napovedi) zdaj in pri prejšnjem zagonu s 14 dnevi:

- **Učinek zakasnitve:** ali se rdeča črta premakne v desno? (Ali model ugotovi, da je vroče, *šele potem*, ko je temperatura že narasla?)
- **Natančnost pri vrhovih:** ali model ne doseže najvišjih temperaturnih vrhov (torej podcenjuje vročino)?
- **Hipoteza:** vročinski valovi se ne pojavijo v trenutku. Zakaj mora UI za njihovo napoved upoštevati zadnjih 14 dni in ne le zadnjih 3 dni?

3. naloga (⚠️ napredna in neobvezna naloga): obrnitev scenarija (preizkus »zimskega anticiklona«)

Ozadje: podnebne spremembe ne prinašajo le vročinskih valov, temveč tudi skrajne ohladitve (npr. vdore polarnega vrtinca). Trenutni sistem je za mraz »slep«, saj išče le visoke temperature.

- **1. korak (potovanje skozi čas):** v **bloku 1** spremenite DISPLAY_FORECAST_START_DATE in DISPLAY_FORECAST_END_DATE na znano hladno zimsko obdobje (npr. od **"2012-01-15"** do **"2012-02-15"**). Spremenljivki LATITUDE in LONGITUDE nastavite na severno mesto, znano po ostrih zimah (npr. **Helsinki**, **Oslo** ali **Winnipeg**). *Namig: koordinate lahko poiščete v Google Maps.*
- **2. korak (pragovi):** v **bloku 1** spremenite CRITICAL_EMERGENCY na vrednost pod lediščem, na primer **-15.0** (°C).
- **3. korak (obrnitev logike):** pojdite v **blok 7** (sistem odločanja).
    - Poiščite vrstico kode: if forecast_temp >= CRITICAL_EMERGENCY:
    - **Dejanje:** operator >= (večje ali enako) zamenjajte z <= (manjše ali enako).
    - *Opomba:* sistem zdaj sproži opozorilo samo, ko temperatura *pade pod* mejo zmrzovanja.
- **Zagon:** izvedite blok 7.

Vprašanje: ali sistem zdaj najhladnejše dni pravilno označi kot »CRITICAL ALERTS«? Razpravljajte, zakaj je prilagajanje logike UI pri spoprijemanju z različnimi vrstami podnebnih tveganj prav tako pomembno kot kakovostni podatki.

4. naloga: razprava

Rezultate nalog 1–3 razložite v okoljskem kontekstu. Na podlagi pogostosti in razporeditve opozoril (blok 7, blok 8) predlagajte ustrezno politiko urbanističnega načrtovanja za ublažitev ugotovljenih tveganj.

Preverjanje znanja

Kaj je glavni namen uporabe modela LSTM (dolgi kratkoročni spomin) namesto preprostega izračuna povprečja pri napovedovanju vročinskih valov?

A. Zgladiti dnevna nihanja in ustvariti pregleden letni graf temperature.
B. Prepoznati tehnične napake in šum v neobdelanih podatkih senzorjev.
C. Zajeti zapletene časovne vzorce in »učinek kopičenja« vročine skozi čas.
D. Odpraviti potrebo po fizičnih vremenskih postajah na oddaljenih območjih.

Katera je najverjetnejša negativna posledica skrajšanja »spomina« UI (Lookback Days) s 14 dni na samo 3 dni?

A. Model bo postal občutljivejši na šum in bo pogosto sprožal lažna opozorila.
B. Model bo trpel zaradi »učinka zakasnitve« in bo vročinski val zaznal šele, ko bo temperatura že dosegla vrh.
C. Model bo samodejno preklopil z globokega učenja na linearno regresijo.
D. Modela ne bo mogoče zagnati, ker bo podatkovni nabor postal premajhen za obdelavo.

Kaj je pri okoljskem napovedovanju glavno tveganje prekomerno prilagojenega modela globokega učenja?

A. Ustvarja preveč posplošene napovedi in ne upošteva lokalnih vremenskih posebnosti.
B. Zapomni si posamezne zgodovinske dogodke (šum) in ne zmore natančno napovedovati ob pojavu novih vremenskih vzorcev, ki jih prej še ni videl.
C. Potrebuje bistveno več računske moči, kot je omogoča strojna oprema.
D. Ne more obdelovati nelinearnih podatkov, kot sta vlažnost ali hitrost vetra.

Kaj je z vidika upravljanja mesta glavna strateška prednost uporabe tega napovednega sistema UI v primerjavi s tradicionalnim »reaktivnim« pristopom?

A. Omogoča, da se viri za ukrepanje ob izrednih razmerah (npr. centri za ohlajanje, ekipe za popravilo omrežja) aktivirajo več dni pred nastopom krize, s čimer se zmanjša škoda za prebivalstvo.
B. Zagotavlja 100-odstotno točnost in v celoti odpravlja možnost lažnih opozoril ali spregledanih dogodkov.
C. Odpravlja potrebo po človeškem odločanju in omogoča, da mesto brez nadzora v celoti upravljajo računalniški algoritmi.
D. Z optimiziranjem semaforjev samodejno zniža dejansko temperaturo mesta.

Kateri pojav najbolje pojasni, zakaj bi lahko model UI za mestno središče z veliko betona napovedal bistveno višje temperature kot za bližnji gozd, čeprav sta si območji geografsko blizu?

A. Učinek urbanega toplotnega otoka (UHI), pri katerem umetne površine zadržujejo in ponovno oddajajo več toplote kot naravno rastlinstvo.
B. Lokalna anomalija zračnega tlaka, ki pogosto nastane nad visokimi stavbami.
C. Pristranskost pri umerjanju satelitskih senzorjev, zaradi katere je odboj od steklenih površin precenjen.
D. »Učinek zakasnitve«, ki ga povzroča velika gostota prometa in zaradi katerega je pretok zraka počasnejši.

Literatura / dodatno branje

[1] F. Chollet, Deep Learning with Python, 2. izd. Shelter Island, NY, ZDA: Manning Publications, 2021 [splet]. Dostopno na: https://sourestdeeds.github.io/pdf/Deep%20Learning%20with%20Python.pdf, dostopano: 15. dec. 2025.

[2] A. Zhang, Z. C. Lipton, M. Li in A. J. Smola, *Dive into Deep Learning*. arXiv:2106.11342, 2021. [splet]. Dostopno na: https://arxiv.org/abs/2106.11342, dostopano: 15. dec. 2025.

[3] Goodfellow, Y. Bengio in A. Courville, Deep Learning. Cambridge, MA, ZDA: MIT Press, 2016 [splet]. Dostopno na: http://www.deeplearningbook.org, dostopano: 15. dec. 2025

[4] J. Heaton, *Applications of Deep Neural Networks*. arXiv:2009.05673, 2020. [splet]. Dostopno na: https://arxiv.org/abs/2009.05673, dostopano: 15. dec. 2025.

[5] Bilal Hassan, Areg Karapetyan, Aaron Chung Hin Chow in Samer Madanat. Climate Adaptation-Aware Flood Prediction for Coastal Cities Using Deep Learning. arXiv:2510.26017 [cs.CV], 2025 [splet]. Dostopno na: https://arxiv.org/abs/2510.26017, dostopano: 15. dec. 2025

[6] Qianqian Zhou, Shuai Teng, Zuxiang Situ, Xiaoting Liao, Junman Feng, Gongfa Chen, Jianliang Zhang in Zonglei Lu. A deep-learning-technique-based data-driven model for accurate and rapid flood predictions in temporal and spatial dimensions. Hydrol. Earth Syst. Sci., 27, 1791–1808, 2023 [splet]. Dostopno na: https://hess.copernicus.org/articles/27/1791/2023/hess-27-1791-2023.pdf , dostopano: 15. dec. 2025

[7] M. Nasourinia in K. Passi, »Wildfire Prediction in British Columbia Using Machine Learning and Deep Learning Models: A Data-Driven Framework«, *Big Data and Cognitive Computing*, let. 9, št. 11, str. 290, 2025. [splet]. Dostopno na: https://doi.org/10.3390/bdcc9110290, dostopano: 15. dec. 2025.

[8] Hyunglok Seo in Yeonjoo Kim. Enhancing Alaskan wildfire prediction and carbon flux estimation: a two-stage deep learning approach within a process-based model. *Environ. Res. Lett.*, 19, 124040, 2024 [splet]. Dostopno na: https://iopscience.iop.org/article/10.1088/1748-9326/ad8bdc, dostopano: 15. dec. 2025

[9] Babak Mohammadi, Mohammed Abdallah, Rachid Oucheikh, Okan Mert Katipoğlu in Majid Cheraghalizadeh. Enhancing streamflow drought prediction: integrating wavelet decomposition with deep learning and quantile regression neural network models. *Earth Sci. Inform.*, 18, 232, 2025 [splet]. Dostopno na: https://link.springer.com/article/10.1007/s12145-025-01736-w, dostopano: 15. dec. 2025

[10] Elisa Felsche in Ralf Ludwig. Applying machine learning for drought prediction in a perfect model framework using data from a large ensemble of climate simulations. *Nat. Hazards Earth Syst. Sci.*, 21, 3679–3691, 2021 [splet]. Dostopno na: https://nhess.copernicus.org/articles/21/3679/2021/nhess-21-3679-2021.pdf, dostopano: 15. dec. 2025