

GimnazijaVič

Strojno pisanje pesmi

s C++ in povratno nevronske

Avtor: Ema Leila Grošelj

Mentorja: Klemen Bajec in Edin Saračević

19. 3. 2020, Ljubljana

Povzetek

Maturitetna naloga zajema dva različna pristopa k pisanju haikujev z računalnikom. Problema se loti tako s povezovanjem naključnih besed iz slovarja v C++-u kot tudi s treniranjem preproste povratne nevronske mreže z interneta (tensorflow.org).

Ključne besede

haiku, C++, strojno učenje, nevronske mreže, RNN

Abstract

The aim of this report is to cover two different approaches to writing haikus with a computer. The problem is solved by connecting random words from the dictionary in C++ as well as by training a simple recurrent neural network from tensorflow.org.

Key words

haiku, C++, machine learning, neural networks, RNN

Kazalo

Uvod	3
Haiku	3
Del 1: C++	5
Kratek opis	5
C++	5
Reševanje problema.....	6
Testiranje programa	7
Zaključek.....	8
Del 2: RNN	9
Strojno učenje.....	9
Nevronske mreže	10
Reševanje problema.....	12
Ugotovitve.....	14
Zaključek.....	15
Zaključek	17
Literatura	18

Uvod

Projektna naloga zajema dva pristopa k reševanju istega problema, kako strojno napisati pesem.

Prvi pristop se težave loti od spodaj, in sicer z nivoja besed. To zahteva bazo besed z njihovimi skladenjskimi lastnostmi in program, ki ve kako besede ustrezno pregibati. Ta metoda je v veliki meri odvisna od naključja, saj temelji na poljubni izbiri besed in na upanju, da bo kateri od izhodnih izdelkov imel, kakšen človeku razumljiv pomen.

Drugi pristop svojega znanja ne črpa iz programerjevega poznavanja jezika, temveč se uči iz baze pesmi, ki že imajo neko bistvo, vsebino. Program v besedilu išče ponavljajoče se vzorce, fraze, pri čemer pa se pojavi problem plagiatorstva. Kje je meja med »copy-pastom« in novim avtorskim delom?

Za osredni cilj sem izbrala haiku, kot eno najkrajših pesniških vrst, ki tolerira le sedanjik.

Haiku

Haiku je vrsta japonske poezije, ki orisuje slike narave oz. želi kar se da na kratek in prepričljiv način predstaviti nek dogodek, situacijo ali misel. Neglede na to, ali gre za opis žive ali nežive narave, bo vedno obstajala neka intuitivna povezava s človeškim razpoloženjem. (Haiku Society of America, 2004)

Sestavljen je iz zelo premišljenega vzorca treh vrstic s po 5-7-5 zlogi (to je skupaj 17), kar se v modernem haikuju počasi opušča. Nekateri prevajalci iz japonščine v angleščino poročajo, da sedemnajst zlogov v japonščini povprečno ustreza dvanajstim zlogom v angleščini. (Haiku Society of America, 2004)

Tradicionalni japonski haiku vsebuje tudi kigo t.j. besedo ali frazo, po kateri lahko ugibamo o letnem času, in kireji t.j. neke vrste govorna punktuacija, ki ustvari premor ali pa poudarek. S tem dosežemo, da je pesem bolj razgibana. Lahko jo nadomestimo z pomišljajem. (Haiku Society of America, 2004)

Rima ni ustrezna pesniška figura, medtem ko je uporaba sedanjika in tretje osebe zaželena. Večina haikujev je brez naslova, metafor in simbolov, ampak nastaja vedno več vrst pseudohaikujev, s katerimi rušimo vsa ta pravila.

Del 1: C++

Kratek opis

Kot vhod program vzame le datoteko sskj2.csv, ki mu predstavlja vir besedišča. Iz dokumenta črpa podatke - tako besede kot tudi njihove lastnosti, kot so roditeljska končnica, spol, končnica v prvi osebi ednine itd.

Program nam vrne niz naključnih besed, ki se v vrsti ujemajo z vektorjem oblika. Vektor oblika je zaenkrat zapisan znotraj programa, in sicer:

```
vector<Vrsta> oblika;
    oblika.push_back(pridevnik);
    oblika.push_back(samostalnik);
    oblika.push_back(glagol);
    oblika.push_back(pridevnik);
    oblika.push_back(samostalnik);
```

V prihodnosti pa si želim, da bi uporabnik lahko sam podal neko zaporedje besednih vrst. Na le-tem pa bi program osnoval pesem. Nekatere oblike kot je na primer glagol, pridevnik, samostalnik so podprte že sedaj, vendar niso v celoti testirane. Pri načrtovanju novih modelov je edini problem v tem, da se besede v sklonu, številu, spolu ipd. zgledujejo po sosedah.

Program vrne vse do deset haikujev, glede na to koliko zlogov imajo skupaj vse besede. Vendar več na to temo sledi v odstavku Reševanje problema.

C++

C++ je splošno namenski, objektno orientiran programski jezik. Pravzaprav gre za razširitev starejšega programskega jezika C, razvil pa ga je Bjarne Stroustrup. (Techopedia)

Še ena nadpomenka zanj bi bil jezik intermediarne ravni, saj obsega značilnosti tako visokih kot tudi nizkih programskih jezikov. Sprva se je imenoval »C z razredi«, saj je imel vse lastnosti jezika C, poleg tega pa je vključeval tudi koncept razredov. Leta 1983 se je preimenoval v C++. Gre za enega najpopularnejših programskih jezikov. Primarno se uporablja za sistemsko programsko opremo, aplikacije... (Techopedia)

Glavni poudarek C++-a predstavlja zbirka prej definiranih razredov. Le-ti lahko potem sprejmejo razne članske funkcije. S tem dosežemo določeno uporabnost. S pomočjo razreda lahko nato ob zagonu programa definiramo več objektov. Poleg tega lahko nek razred (subclass ali podrazred) deduje od nekega drugega razreda (superclassa ali nadrazreda). Deduje se javni in zaščiten del razreda, ne pa tudi zasebni. (Techopedia)

Podpira razne operatorje, kot so operatorji za primerjavo, aritmetiko, manipulacijo z biti in logični operatorji ter tudi preobremenjevanje. Glavni koncepti C++-a so polimorfizem, virtualne in prijateljske funkcije, šablone, imenski prostori in kazalniki. (Techopedia)

Sama sem v programu uporabila števila, nize, vektorje, funkcije, enum, razrede, dedovanje, virtualne funkcije, branje iz dokumenta, if stavke, for zanke...

Reševanje problema

Reševanja problema sem se lotila po več sklopih.

Prvi sklop predstavlja pretvarjanje vsake vrstice sskj2.csv datoteke v primeren subclass (Samostalnik, Pridevnik, Glagol, Prislov, Predlog ali Veznik) superclassa Beseda. Objekte istega razreda spravimo v skupen vektor.

```
vector<Pridevnik> pridevniki;  
vector<Samostalnik> samostalniki;  
vector<Glagol> glagoli;  
vector<Prislov> prislovi;  
vector<Veznik> vezniki;
```

V okviru drugega sklopa ustvarimo vektor kazalnikov na Besede *haiku* in vanj glede na obliko te kazalnike na naključne elemente tudi dodamo. Besede na katere le-ti sedaj kažejo so še v osnovni obliki. Zato v tretjem sklopu na njih kličemo starševsko funkcijo set().

Le-ta je najpomembnejša funkcija mojega programa. Gre za virtualno funkcijo razreda Beseda, ki jo od nje dedujejo podrazredi Pridevnik, Glagol in Samostalnik. Vsak se nastavi drugače, vsi pa za to potrebujejo kazalnik na vektor *haiku* in svoj index v le-tem. Po tem lahko v odvisnosti od drugih besed nastavijo svoje parametre. Nekateri taki so spol, število, sklon itn. glede na to za katero besedno vrsto gre. Te podatke funkcija nato uporabi za pregibanje besede. Kar znotraj vsake od funkcij se nahajajo vektorji, vektorjev, ... , nizov, v katerih so shranjene tipične končnice za posamezne besedne vrste. Končnica je odvisna od končnice

temeljne besede, od končnice uvožene iz datoteke sskj2.csv, sklona, spola in števila. Tako imamo pri set() funkciji pri samostalniki na primer seznam seznamov seznamov nizov *vzor*:

```
vector<vector<vector<string>>> vzor = {
/*0*/({{"", "a", "u", "", "u", "om"}, {"a", "ov", "oma", "a", "ih", "oma"}, {"i", "ov", "o
m", "e", "ih", "i"}},
/*1*/({{"", "a", "u", "", "u", "em"}, {"a", "ev", "ema", "a", "ih", "ema"}, {"i", "ev", "e
m", "e", "ih", "i"}}, /*če živo z a-jem*/
/*2*/({{"a", "e", "i", "o", "i", "o"}, {"i", "", "oma", "i", "ah", "ama"},
{"e", "", "am", "e", "ah", "ami"}},
/*3*/({{"a", "e", "i", "o", "i", "o"}, {"i", "", "ama", "i", "ah", "ama"},
{"e", "", "am", "e", "ah", "ami"}},
/*4*/({{"", "i", "i", "", "i", "jo"}, {"i", "i", "ma", "ti", "ih", "ma"},
{"i", "i", "im", "i", "ih", "mi"}},
/*5*/({{"o", "a", "u", "o", "u", "om"}, {"i", "", "oma", "i", "ih", "oma"},
{"a", "", "om", "a", "ih", "u"}},
/*6*/({{"e", "a", "u", "e", "u", "em"}, {"i", "", "ema", "i", "ih", "ema"},
{"a", "", "em", "a", "ih", "u"}},
/*7*/({{"ev", "ve", "vi", "vo", "vi", "vijo"}, {"vi", "ev", "vama", "vi",
"vah", "vama"}, {"ve", "v", "vam", "ve", "vah", "vami"}},
};
```

Člansko funkcijo kličemo v posebnem vrstnem redu. Najprej jo kličemo na glagolih, nato na samostalniki itn., saj se samostalniki na primer v številu ujemajo z glagoli, v sklonu pa se po njih ravnaajo glede na pozicijo pred ali za glagolom. Ta del sem nekoliko poenostavila, saj program vsakemu samostalniki pred glagolom z aktivnim statusom priredi imenovalnik, tistim za njim pa tožilnik. V načrtu je bila še izjema za povratne glagole, katerim bi naključno sledili samostalniki v petem ali šestem sklonu. V tem primeru samostalniki spredaj dodamo tudi predlog.

Nato »pesem« sprintamo, pri čemer se poslužujemo preobremenitve (overloada) operatorjev. Natisnemo pa še zdaleč ne vseh primerkov, temveč le tiste, katerih besede imajo skupaj med petnajst in dvajset samoglasnikov. Po številu samoglasnikov sklepamo o številu zlogov. Z novo vrstico začnemo vsakič, kadar število zlogov v verzju preseže štiri.

Testiranje programa

Program sem testirala tako, da sem za najosnovnejše oblike preverjala pravilnost spreganja in sklanjanja. In ga sproti prilagajala za nekatere izjemne skupine tega prečudovitega slovenskega jezika. Nekaj težav so mi povzročili akcenti v prvotnem sskj.csv, ki sem jih odstranila s pomočjo najdi in zamenjaj.

S šumniki ni težav če le ne nastopajo na kočljivih mestih v končnicah. Probleme so mi povzročali tudi, ko sem želela pri samostalniki preveriti prisotnost črk c, j, č, ž ali š na koncu besede. To je pomembno, saj le-te črke povzročijo preglas predležeče črke o v črko e. Velja pa le za nekatere sklone srednjega in moškega sklona.

Na primer: mesto => mestom ampak učenec => učencem

Poleg tega problem predstavlja tožilnik prve moške sklanjatve. V primeru, da samostalnik pomeni živo bitje se beseda podaljša za »-a«. V nasprotnem primeru je končnica ničta. Program pa med živim in neživim žal še ne loči.

Zaključek

Program je nepopolnosti navkljub rešil moj problem. Poleg tega sem utrdila osnovno znanje C++-a in se naučila nekaj novih pristopov k programiranju. Prepričala sem se tudi, da smisel leži tudi v stvareh, ki niso načrtno smiselne. Treba ga je le poiskati. Prednost take poezije je, da jo vsak lahko razume drugače. Res pa je, da zahteva določeno mero angažiranosti. Kot je razvidno iz spodnjih primerov je računalnik zmožen pisati zabavno, impresionistično in celo osebno izpovedno poezijo.

lasulja prelistava marec
šesdesetletnika štejeta prirednost
umirjanja zapičijo dvigalo

preskrbna egoizma
zaživita švigljasto
gorljivost

trifazni šerpi
odstreta neperspektivn(ega)
piton(a)

štafažen krempelj
oznoji preresničen
tretjeletnik

mnogolistni položniki
ogrinjajo prepolno
obalo

zamrzovalen
zapetež utrjuje
laksno zvezdišče

ŠVIGLJAST = zelo velik in suh

ŠTAFABA = nebistveni del fotografije, slike za ozadje, dopolnitev glavnega motiva

LAKSEN = ki ni natančen, strog, zlasti glede disciplinskih, moralnih načel

Del 2: RNN

Strojno učenje

Strojno učenje je podvrsta umetne inteligence, ki sistemom omogoča avtomatično učenje iz preteklih izkušenj in izboljševanje s pomočjo le-teh brez neposrednega programiranja. Učenje sestoji iz obdelovanja podatkov, pa naj gre za dražljaje iz okolja, zanj sestavljene učne primere ali navodila, pri čemer v podatkih išče vzorce, ugotovitve pa uporabi za pridobivanje točnejših rezultatov. Cilj ostaja, da se računalnik uči sam, brez človeškega vmešavanja, in naučenemu prilagaja svoje delovanje. (expertsystem.com, 2017)

Učenje glede na pristop razdelimo v 3 skupine. Algoritme pri nadzorovanem učenju se trenira na vadbenih podatkih oz. učni množici primerov, ki so največkrat sestavljeni iz vhoda in pričakovanega izhoda. Po tej poti dobimo funkcijo ali napovedni model, s katerim lahko predvidevamo izhode tudi pri vhodih, za katere prej nismo imeli podatka. Učinkovitost učenja izmerimo tako, da za nekaj vadbenih primerov, ne da bi se algoritem prej na njih učil, oz. za primere iz validacijske množice pridobimo izhodno vrednost in jo primerjamo s podano. Predstavnik tega pristopa je klasifikacija, pod katere okrilje spadajo metode, kakršne so odločitvena drevesa, bližnji sosed, diskriminantne funkcije, hibridni algoritmi, umetne nevronske mreže in Bayesove mreže. (expertsystem.com, 2017) (Zacirkovnik, 2013)

Nasprotno se algoritmi tipa nenadzorovanega učenja učijo iz neurejenih podatkov in jih hkrati tudi glede na kriterij razdelijo v več različnih kategorij. Iščejo funkcijo, ki kar najbolje opiše skrite vzorce, ne da pa nujno izhodnih podatkov. Rezultat takšnega učenja je povzetek učne množice primerov. Primer je tako regresija (regresijska drevesa, linearna regresija, lokalno obtežena regresija, metoda podpornih vektorjev v regresiji) kot tudi klasifikacija. Delno nadzorovano učenje, navadno sestavljeno iz malo urejenih in veliko neurejenih podatkov, da v primerjavi z nenadzorovanim učenjem nekoliko točnejše rezultate. (expertsystem.com, 2017) (Zacirkovnik, 2013)

Spodbujevalno učenje sloni na principu nagrajevanja in kaznovanja. Sistem pri različnih nastavitvah pridobiva povratne informacije, išče parametre, ki dajo ciljne rezultate. Ta način se uporablja za reševanje problemom kontrole dinamičnih sistemov, raznih optimizacijskih problemov in pri igranju iger. (expertsystem.com, 2017) (Zacirkovnik, 2013)

Izbora metode se lotimo na sledeč način. V primeru, da nimamo ciljne vrednosti, izberemo nenadzorovano učenje. Če jo imamo in je le-ta diskretna (da/ne, A/B), posežemo po nadzorovanem. Številčna ciljna vrednost implicira uporabo regresije. (Zacirkovnik, 2013)

Strojno učenje omogoča obdelavo ogromnih količin podatkov, pri čemer je veliko zmogljivejši kot človek, saj je hitrejši in natančnejši. Uporabljajo ga za analizo podatkov, podatkovno rudarjenje, gradnjo numeričnih, kvalitativnih modelov itn. Lahko bi rekli, da strojno učenje spreminja podatke v informacije. Gre za močno interdisciplinarno znanost s temelji v računalništvu, statistiki in inženirstvu. Prav povezava s statistiko je vredna poudarka. Medtem ko se statistika ukvarja z vsem od postavljanja hipotez do testiranja, se strojno učenje bolj posveča oblikovanju postopkov. (Zacirkovnik, 2013)

Nevronske mreže

Tehnološki začetki nevronske mreže segajo v zgodnja 40. leta, ko sta McCulloch in Pitts postavila matematični model živčne celice. Rosenblatt je leta 1957 razvil perceptron in ga uporabil za prepoznavo ročno napisanih števil. Teoretična zasnova mrež se skozi čas ni dosti spreminjala. (Zacirkovnik, 2013)

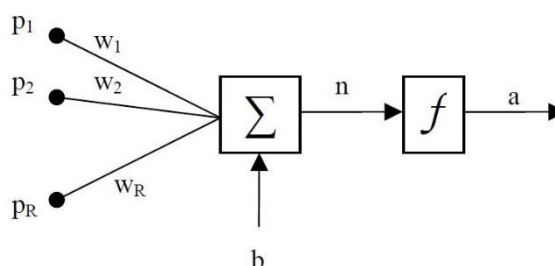
Nevronske mreže so sistem umetne inteligence, ki posnema naravno inteligenco. Od biološke osnove so umetni nevroni prevzeli principe, kot so številni dendriti oz. vhodi, sinapse do drugih nevronov in prag, katerega je treba preseči, da gre signal sploh naprej. (Lukšič, 2003)

Sposobnosti posameznega nevrona so torej sprejemanje vhodnih signalov, njihova ustrezna obravnava in spreminjanje ter oddajanje. Matematično nevron definiramo kot matematično funkcijo:

$$a = f\left(\sum_{i=1}^n w_i p_i + b\right)$$

pri kateri so:

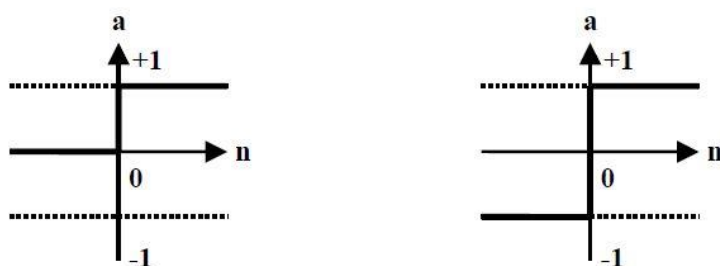
i – število vhodov
 p_i – signal i-tega vhoda
 w_i – utež i-tega vhoda
 b – konstantni odmik nevrona ali vrednost pristranskosti
 f – transformacijska ali aktivacijska funkcija
 a – izhodni signal
 (Lukšič, 2003)



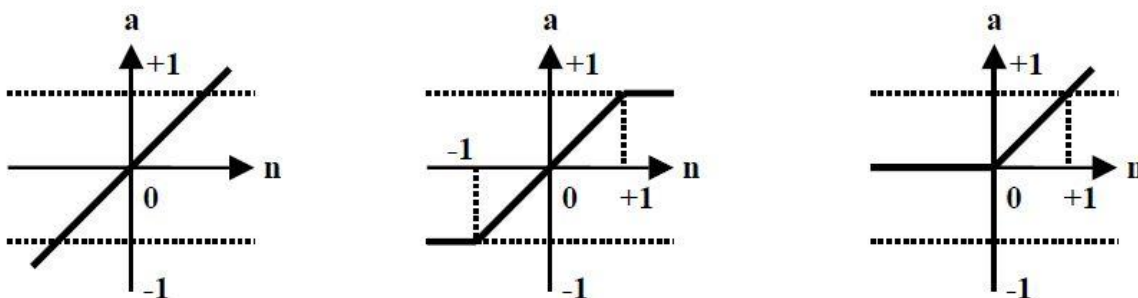
Nevron učimo tako, da skozenj poženemo vhodne podatke in nato izhodni signal primerjamo z želenim. Po potrebi lahko spreminjamo uteži in odmik. To dosežemo z odvodom, ki predstavlja spremembo uteži glede na spremembo napake. (Zidarn, 2019)

Poznamo več transformacijskih funkcij:

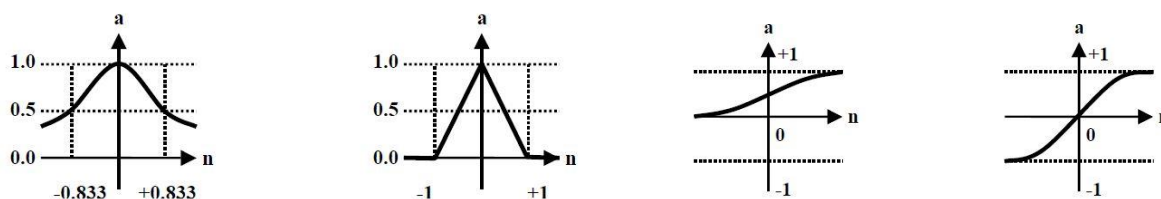
Stopničasta (binarna) funkcija za negativne vrednosti vrne 0 za nenegativne pa 1. Gre za zelo pogosto uporabljeno funkcijo, predvsem pri sistemih za odločanje, kjer se izhod glasi bodisi da bodisi ne. Simetrično stopničasta (bipolarna) funkcija je enaka stopničasti, razen tega da pri negativnih vrednostih vrne -1 namesto 0. (Lukšič, 2003)



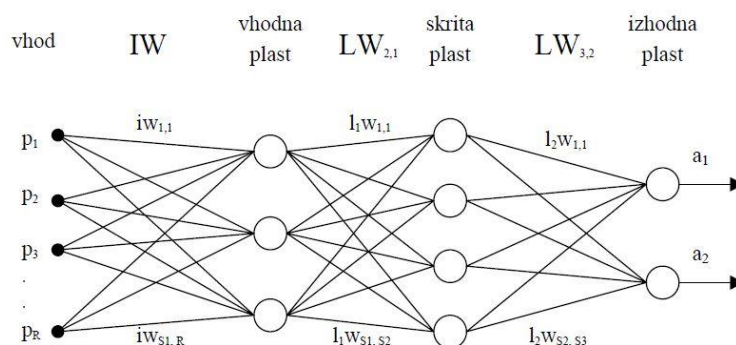
Linearna funkcija je ena najpogostejše uporabljenih v izhodni plasti mreže, kadar pričakujemo izhode poljubne velikosti. Omejeno linearna funkcija linearno preslika le vrednosti med -1 in 1. Vrednosti manjše od -1 omeji na to vrednost, tako kot za vrednosti večje od 1. Uporabimo jo, kadar želimo drugače linearen izhod omejiti na neko zalogo vrednosti. Pozitivno (negativno) linearna funkcija je funkcija, ki ima izhode omejene le na pozitivne (negativne) vrednosti in nič. (Lukšič, 2003)



Obstajajo še mnoge druge, kot so radialna po formuli normalne porazdelitve, trikotna, sigmoidna in hiperbolično tangenta funkcija. (Lukšič, 2003)



Glede na število plasti in nevronov, povezanost (stopnja, smer) in izbor transformacijskih funkcij ločimo razne tipe nevronske mreže. Najbolj znane so enostavni perceptron, večplastni perceptron, radialno zasnovane mreže, samo-organizirajoče se mreže in povratne mreže. Perceptron je najosnovnejša oblika, ki praviloma vsebuje stopničasto funkcijo in opravlja bolj enostavne naloge. Večplastni perceptron že vsebuje bolj kompleksne transformacijske funkcije. Taka mreža vsebuje že vsaj eno skrito plast, kot taka pa se uporablja za napovedovanje časovnih vrst in aproksimacijo funkcij. (Lukšič, 2003)



Če se posvetimo še povratnim mrežam; povratnost je v tem, da se ista funkcija izvede na vseh vhodnih podatkih, medtem ko so izhodne vrednosti odvisne od prejšnjih vhodnih podatkov, in sicer na način, da se informacija o starejših podatkih ohrani v skritih plasteh. LSTM (Long Short-Term Memory) mreže si še lažje predvsem pa dlje zapomnijo pretekla stanja. Celica ima t.i. vrata pozabe, s katerimi uravnava, koliko informacije si se vredno zapomniti oz. posredovati naprej. GRU mreža je le poenostavljena LSTM mreža. (Mittal, 2019) (Zidarn, 2019)

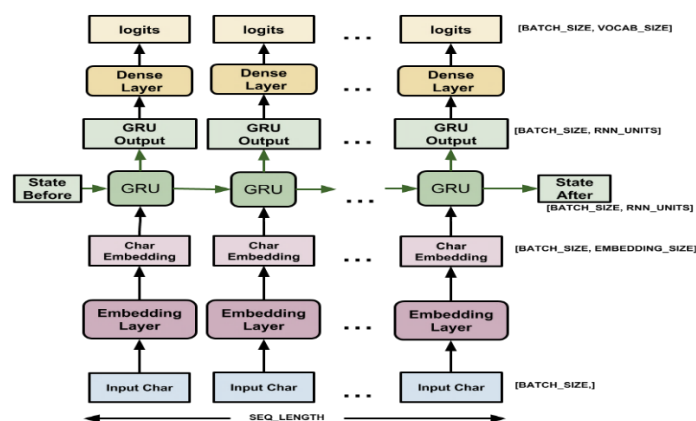
Reševanje problema

Zaradi premalo predznanja v tensorflowu in kerasu sem se odločila, da bom za potrebe reševanja problema priredila že delujočo objavljeno kodo. Ta koda je bila prvotno namenjena Shakespearju in je objavljena na tensorflow.org. (Tensorflow)

Največ časa mi je vzelo urejanje učnega materiala in sicer nagrajenih haikujev in haikujev že uveljavljenih avtorjev. Ves učni material mi je v elektronski obliki priskrbel profesor Edin Saračević. Kot sem že rekla, sem jih nato spravila v obliko: brez avtorjevega imena in prazna vrstica za ločnik.

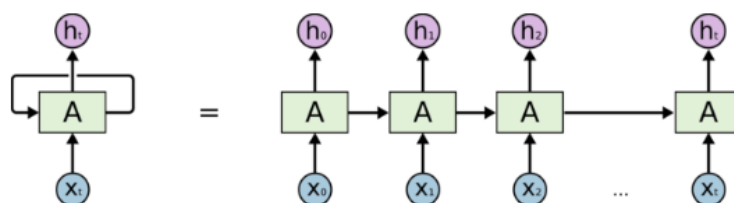
Koda za temelj vzame znak (črko, številko, presledek, ločilo, odstavek). Vsakemu različnemu znaku priredi številko, torej ga numerično reprezentira. Cilj treninga mreže je, da nam vrne najbolj statistično verjeten zank, ki bi sledil nekemu zaporedju znakov. S tem namenom naredimo vadbeno množico sestavljeno iz parov; niza 100 znakov in znaka, ki mu v resnici sledi je pa tudi ciljna vrednost in se uporabi pri preverbi točnosti izračunov.

Sama mreža je sestavljena iz treh plasti: vhodne, skrite (GRU) z 1024 nevroni in izhodne.



Treniramo jo z uporabo optimizatorja – za nas je to Adam, ki s kriterijsko funkcijo računa odstopanja izhodnih podatkov od zaželenih. Poznamo več kriterijskih funkcij: MSE (slov. srednja kvadratna napaka), MAE (slov. srednja absolutna napaka), prečna entropija (angl. Cross-Entropy Error), itn. Med treniranjem se, če je le-to uspešno, točnost s časom izboljšuje. Ta čas lahko določimo z določitvijo števila t.i. epoh; zgornjo mejo ponovitev učnih prehodov podatkovne množice. (Zidarn, 2019)

Besedilo generiramo, začeniši z poljubnim začetnim nizom znakov kot vhodnim podatkom. Določimo tudi število znakov, ki jih želimo sproducirati.



Z nastavitvijo t.i. temperature vplivamo na kategorično distribucijo in lahko vplivamo na stopnjo naključnosti v našem rezultatu.

Ugotovitve

Prvotni učni material je sestavljalo komaj 500 haikujev, v zadnji verziji sem jih uspela zbrati 8 300. Seveda se razlika odločno pozna, saj so bili prvotni izhodni haikuji komaj kdaj različni od primerkov iz vhoda (angl. overfitting). S po-16-terjenjem so prišli tudi nekateri inovativnejši hakuji, med njimi celo kakšen tak s samimi edinstvenimi verzi in celo globokim sporočilom. Kar pa se tiče časa učenja sem prišla do zaključka, da treniranje dlje kot 100 epoh ni smiselno, saj se mreža od takrat naprej ne izboljšuje več.

Primer 1:

ново leto
kaj imam
samo sveža bolezen

po:

ново leto
kaj imam
samo tebe
(Katja Čokl)

aktualen

Primer 2:

Sreča je. (5 pojavitev)
Jesenski veter... (7 pojavitev)
celo večnost (2 pojavitvi)

najbolje potrdi, da gre pri mrežah pravzaprav le za statistično obdelavo

Primer 3:

zvonec zazvoni
 tiho tiho mojega
 telesa prepot

nova beseda

Primer 4:

danes pazi,
 počitek na nebu,
 sveti na telo.

popolnoma edinstven

Še nekateri in nekatere misli:

Kapljice drobne
 padajoče listje pa
 kaplje dežja

Učilnice so,
 uživam ta svet odpada

mreža v laseh
 svetilka snežinka,
 na deževen dan

Profesor besed
 ničesar ne naju
 daleč od hrušk

a kje sem in kdaj sonce

leti ledena,
 želje kapljajo
 težka dolgčas na cesti.

Diham vojni greh.

Maček pred mano
 gre hiša brez besed.

Tudi mlade besede ni več.

Zaključek

Zavedeti se moramo, da je število haikujev v slovenskem jeziku, sploh v elektronski obliki omejeno. Nekaj sto sem jih celo prepisala vendar je to mukotrpno delo, saj je treba število haikujev, da so spremembe sploh vidne, vsaj podvojiti.

Zanimiva se mi zdi tematika pesmi, ki se je v veliki meri obdržala. Gre za temo narave, ljubezni, obstoja in seveda šole.

Omeniti je treba še človeški faktor, ki je žal neizogiben, saj je izhod izgledal nekako takole:

500 haikujev 10 epoh	1000 haikujev 10 epoh	1000 haikujev 50 epoh	1000 haikujev 100 epoh	1000 haikujev 500 epoh	1500 haikujev 100 epoh	8300 haikujev 100 epoh
ROMEO: upere ba j gabojcopect k a.	ROMEO: armjs ejaag ŽEotz ltgarnetore ulm pariloz i	haiku tvaj vepovilo lasenj boli.	haiku, pazipam predlo vso. Prela kolečenje sneti samo uri ti te močem.	haiku, začudenje v zraku, skrivnostna želja. Pozimi cveti samo svečka iz nosa – ta hudi prehlad.	haiku – nardo za pretoplo. Veliki obsed mi nese sinice vseje v pozni jeseni, marv snemir, nisim inala.	haiku vsi metulji Avto na dvorišču, zabava na poti, spomine a sanje in on mi je samo v srcu kotičko
1 čnanave temiojecic 3 vlna opo za lpanzelna. žeka.	vedž vp v siedn, nahe. do er kjosetbaa. uma oop, n vtizaknejdso	Danje taguč je zgozi snotim seububistvije, v gljuboze vstri stenin b njubomja – čivse se rka.	Prazni boreča in šel ni spomin. Sam te vemeš, potnica – spotoni brustiča	Ko čas beži, celemu svetu se mudi. Jaz pa v šoli.	Vsi pred tv-jem s šelom okrog vratu, tekma se začne.	Življenje je pot resnice in laži, ki se skriva v duši.
arepujagpas bni,	detmive otv jene .a. m soeij aue juo n m.hba,	Kočno tišve žilo, žipom ze demnapreši	dežev i trojih je vstra naškovniko hutim d MMitre	sedim v šoli vse v meni je hladno sončno je nadoledu svoje trava, potok teče še naprej, srnjaka poglej.	Sovražim zimo, ker je zunaj mrzlo, na svet, novo tem puh.	Zima besede sladkejšega dne.
jehi m Mp i ogegekčrahmmi b.	lvom ui imo .čgoahoi zččoaaj, o tičšesjčš	Tišav m sonco, insu bespoblo.	te ta nečenje, ku od ljubezen blazha je vsi voza vreče kam.	Poti petek zvečer pripravljam se dve uri njega ni tam.	Govorica za srečno zrcalo, vsako jutro mi želja, vonj me šelo sreča, list na mizi čaka, jaz pa vračnja.	Razprostiram po testu, zavejte na moji koči.
. NvamŠ e t b semieha Mse sje lvtnevjgaKvo ss apo . vekaa. Va,utevo noo dten tomipš inoelscčj t ah.oža ie se zni at nbsemad.nitv a Žvoo,	ckaatas i pe jimeei orlee. nagrij džtjdi nži tnhzomioioaeijaLue e stse.	Tisto ja spom. keka demen pi – trak spot, pol solochi seme ozmučna pogi pame – v dreč vemeski naroj, skoj zanaaj? Detesa se vloka spim dene n že, v ure bled ne klazanjes špomlje rasen sen nestevak.	Listle še pričči... enako čnoči oblako brez obladi. Lustje že smeje poleži, dnobke se je ... kot jesens v moči od.	Luč v podhodu pot v neznano kot moje življenj samo, kadar sem žalostna. Hočem špricati pouk, a ne morem, ker me je nekdo zaparkiral.	v otežniku – na belem ovratu – da lahko piščava ubaje, znanje br zsejem uri.	Ljubiti nežno gre na poletem ga, ponazarja ljudi.
oUaš, žbo, amoliubtsc prsrleto i, o tja,ztnjpkereieiodanebe rtu ež tjeđkožsar.	Pcjeaed ve l uke. La o aeksdlojj ten,nd, thjikojuuid vbkajp,lin a enrei lshesroivnts	ovostisam se miram.	Kodli gres napal, bidljo nasmeš.	Uživajmo v	Matematika. Kako močno svetijo smeh me zanari leto,	Zvezki na planot sem preglasi pred samo.

Še vedno je na človeku, da presodi, kaj je dobro in kaj ne.

Seveda ne gre vse krivde zvaliti na (v obsegu in nikakor ne vsebini) skromen učni material. Kriva je tudi mreža, ki bi jo bilo mogoče obogatiti z več in kompleksnejšimi plastmi. Prav tako bi bilo vredno preučiti parametre in njihov vpliv na proizvedeni tekst. Nato bi lahko z merjenjem odličnosti haikujev navigirali k optimalnim parametrom in sestavi mreže.

A kaj ko se odličnost poezije težko oceni s številko. Kako zahtevno je šele k ocenjevanju priučiti algoritem. Prva ovira, ki jo vidim so razlike med slogi različnih avtorjev. Če bi tako na primer računalniku kot dobro prakso predstavili Prešerna, kaj bi porekel, če bi mu pokazali Kosovelov kons. Zdi se, da stroj težko loči med nesmiselnostjo in inovativnostjo; seveda ni edini.

Zaključek

Za bolj inteligentno se je vsekakor izkazala druga rešitev, ta z umetno inteligenco. Ne le, da je dala bolj globoke in človeške izdelke, ima tudi več potenciala z nadaljnjo nadgradnjo.

Literatura

dijaski.net. *dijaski.net/gradivo*. [Elektronski] https://dijaski.net/gradivo/slo_sns_slovnica_03.

dr. Vidovič, Muha in Toporišič, Jože. 2018.

studentski.net/gradivo/ulj_fif_sl1_msk_sno_studijsko_gradivo_2009_2010_01. *studentski.net*. [Elektronski] 2018.

https://studentski.net/gradivo/ulj_fif_sl1_msk_sno_studijsko_gradivo_2009_2010_01.

expertsytem.com. 2017. machine-learning-definition. *expertsytem.com*. [Elektronski] 2017. <https://expertsytem.com/machine-learning-definition/>.

Haiku društvo Slovenije. 1999 - 2006. Letni časi. [Elektronski] 1999 - 2006. <http://www.tempslibres.org/hcs/letni/dhome.html>.

Haiku Society of America. 2004. Official Definitions of Haiku and Related Terms. 2004.

Holmes, Vicki L. in Moulton, Margaret R. 2001. *Writing Simple Poems*. 2001.

Issa. Izbor.

Lukšič, Borut. 2003. Uporaba nevronske mreže za napovedovanje tečajev vrednostnih papirjev. [Elektronski] 2003.

Majcen, Tatjana. 2009. Tatjana Majcen - primorska pesnica. [Elektronski] 2009. <http://www.tatjana-malec.si/default.asp?Id=368>.

Mittal, Aditi. 2019. Understanding RNN and LSTM. *Towards Data Science*. [Elektronski] 2019. <https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e>.

Osti, Josip. 2015. *Majhna pesem*. 2015.

Saračević, Edin. *haikuji*.

slovenski dijaki. 2008-2020. *Haiku zborniki*. 2008-2020.

stackoverflow.com. stackoverflow.com/questions. *stackoverflow.com*. [Elektronski]
<https://stackoverflow.com/>.

Techopedia. www.techopedia.com/definition/26184/c-programming-language. *www.techopedia.com*.
[Elektronski] <https://www.techopedia.com/definition/26184/c-programming-language>.

Tensorflow. Text generation with RNN. [Elektronski]
https://www.tensorflow.org/tutorials/text/text_generation.

Zacirkovnik, Andreja. 2013. Orodja za strojno učenje - klasifikacija. [Elektronski] 2013.
<https://dk.um.si/IzpisGradiva.php?id=42095>.

Zidarn, Rok. 2019. Avtomatsko povzemanje slovenskih besedil z globokimi nevronskimi mrežami.
[Elektronski] 2019. <https://repozitorij.uni-lj.si/Dokument.php?id=125791&lang=slv>.

Vse slike so pridobljene na istih virih kot besedilo okoli njih.