Univerzitet u Beogradu
Elektrotehnički fakultet

LSTM za klasifikaciju cifara sa senzora kretanja

projekat iz veštačke inteligencije

predmetni profesor:
Predrag Tadić

student:
Stefan Petrović, 2017/3359

Beograd, 23. VIII 2018.

Sadržaj

Sažetak

U ovom radu je opisan način kojim se pomoću *LSTM* rekurentne mreže mogu klasifikovati cifre pisane u vazduhu, korišćenjem senzora za merenje kretanja sa 6 stepeni slobode. Sav softver za projekat je moguće naći na sledećem linku https://github.com/stefan-ptrvch/lstm_pencil

Ključne reči: *klasifikacija cifara, LSTM, Keras, mbed, MPU-9150, Nucleo*

1. Opis zadatka

Cilj zadatka je klasifikacija vremenskih sekvenci dobijenih sa akcelerometra i žiroskopa, koje predstavljaju cifre od 0 do 9. Cifre se pišu u vazduhu, pomeranjem sklopa senzora i razvojne ploče u ravni paralelnoj podu. Klasifikacija se obavlja korišćenjem *LSTM* rekurentne neuralne mreže. Sklop mikrokontroler/senzor obavlja akviziciju podataka i prosleđuje ih na računar, gde *LSTM* klasifikuje sekvencu i ispisuje u prozoru o kojoj cifri se radi. Program radi u realnom vremenu.

2. Rešenje zadatka

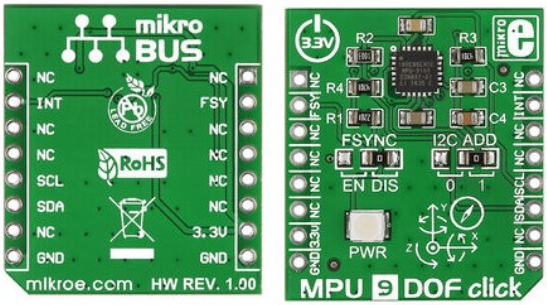
2.1 Opis alata

U nastavku je dat pregled hardverskih i softverskih tehnologija korišćenih za rešavanje navedenog zadatka, kao i kratak opis *LSTM* rekurentnih mreza.

2.1.1 9DOF click

Čip korišćen za merenje kretanja je *MPU-9150 System in Package*, koji sadrži *MPU–6050* senzor (u kome su akcelerometar, žiroskop i digitalni procesor kretanja), kao i *AK8975* digitalni kompas. Digitalni procesor kretanja i digitalni kompas nisu upotrebljeni u ovoj aplikaciji. Svaki od senzora ima 3 stepena slobode, što ukupnom pakovanju daje 9 stepeni slobode (*9DOF*).

Senzor se nalazi na razvojnoj ploči *MPU 9DOF Click* proizvođača *Mikroelektronika*. Ploča podržava *mikroBUS* format za komuniciranje sa mikrokontrolerom i napaja se sa 3.3V. Za iščitavanje podataka korišćen je *I2C* protokol.

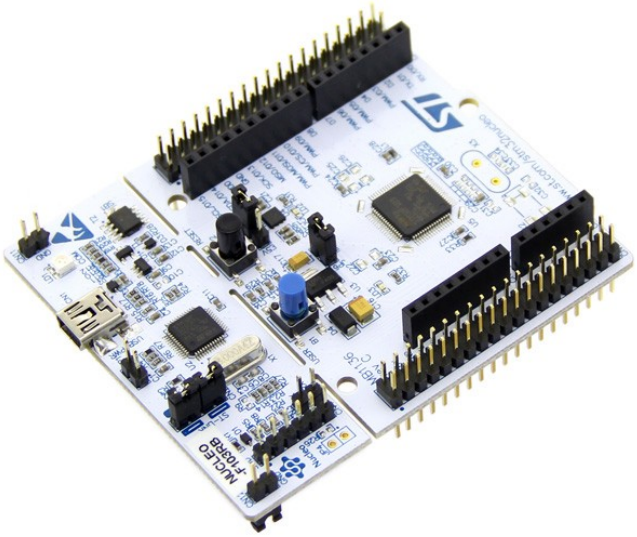


Slika 2.1: 9DOF click

2.1.2 Nucleo F103

Embeded platforma odabrana za realizaciju projekta je *NUCLEO-F103RB*, sa 32-bitnim *ARM Cortex M3* mikrokontrolerom. Platforma je podržana od

strane *mbed* softverskog okruženja, što je dosta olakšalo razvoj embeded dela aplikacije.



Slika 2.2: Nucleo F103RB

2.1.3 mbed

Softversko okruženje odabrano sa razvoj aplikacije na strani mikrokontrolera je *mbed*, obzirom da pruža brzo razvijanje softvera. Koristi objektnu paradigmu u jeziku *C++* što omogućuje dosta jednostavno i intuitivno upravljanje periferijama mikrokontrolera.

2.1.4 Python alati

Za razvoj aplikacije na strani računara, korišćen je *Python* programski jezik, sa bibliotekama *Keras* [1], za obučavanje neuralnih mreža, *scikit-learn* [2] [3], za pomoćne funkcije pri pripremi podataka za obučavanje, *pandas* [4] kao glavna biblioteka za obradu podataka, i *matplotlib* [5] za grafičko prikazivanje rezultata. Pored toga, korišćena je biblioteka *pyserial* za serijsku komunikaciju između mikrokontrolera i računara.

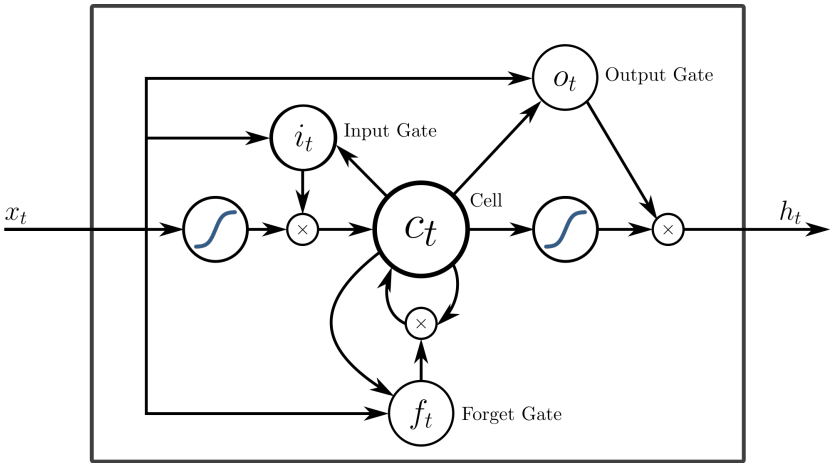
2.1.5 Google Cloud Machine Learning Engine

Obizorm da je obučavanje *LSTM* mreža numerički zahtevno, i obzirom na nedostupnost jake grafičke karte, iskorišćen je *Machine Learning Engine*, *Google Cloud* platforme, preko kog je moguće za svako obučavanje modela automatski

podići predefinisanu *Linux* virtuelnu mašinu (u cloud-u), koja poseduje *NVIDIA Tesla K80* grafičku karticu, pomoću koje obučavanje ide jako brzo. Naplaćivanje usluge se vrši po minutu, ali *Google* nudi 300 USD za sve nove korisnike.

2.1.6 O LSTM

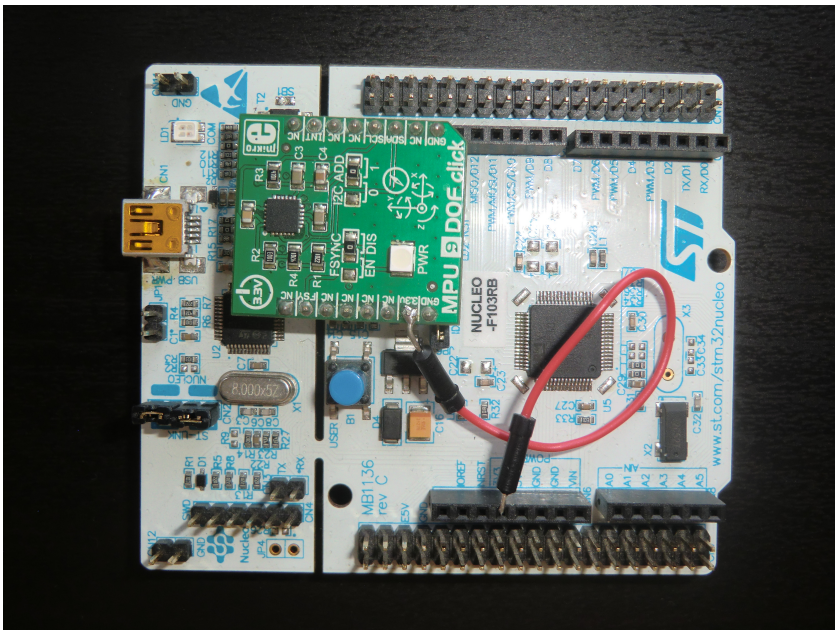
LSTM je tip rekurentne neuralne mreže. Za razliku od običnih neuralnih mreža, rekurentne mreže su u stanju da klasifikuju sekvencu podataka, što ih čini pogodnim za obradu signala sa senzora. Prilikom obučavanja rekurentnih mreža dolazi do problema iščezavajućih gradijenata, što za posledicu ima nemogućnost modelovanja zavisnosti odbiraka koji su u sekvenci daleko jedni od drugih (nemogućnost modelovanja zavisnosti početka i kraja sekvence, ako je sekvenca dovoljno duga). *LSTM* rešava taj problem uvođenjem drugačije arhitekture same mreže [6].



Slika 2.3: Šematski prikaz *LSTM* rekurentne neuralne mreže

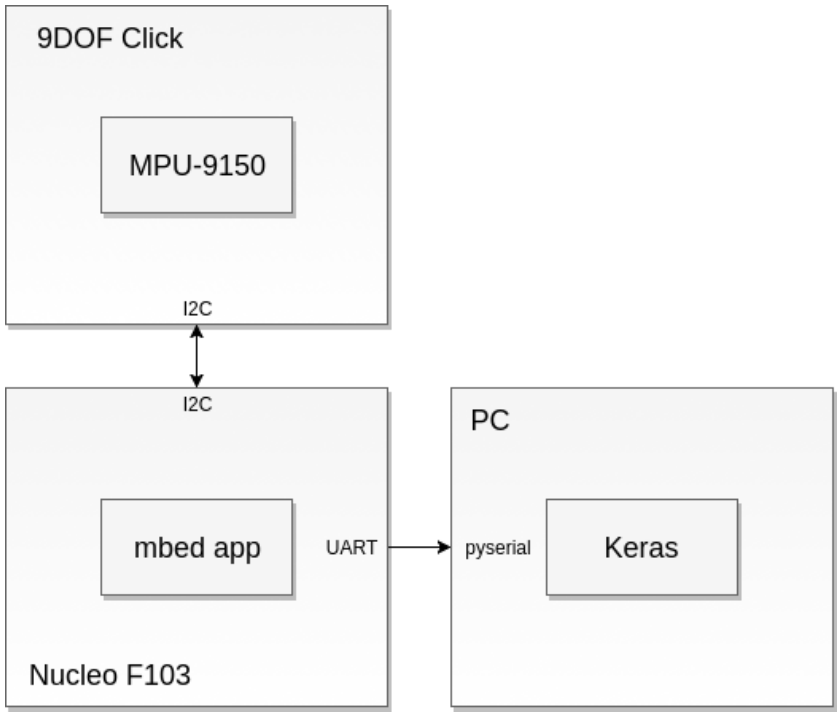
2.2 Povezivanje sistema

Senzorska ploča je povezana sa *Nucleo* pločom preko *mikroBUS* interfejsa, korišćenjem samo pinova za *I2C*, masu i napajanje. Ispostavilo se da je pinout na *Nucleo* ploči takav da dozvoljava direktno "ubadanje" *click* ploče u header razvojne ploče mikrokontrolera, tako da se pinovi za *I2C* i masu poklope, pri čemu je napajanje bilo potrebno dovesti preko kratkospojnika.



Slika 2.4: Povezivanje senzora i razvojne ploče

Između mikrokontrolera i računara postoji serijska komunikacija preko koje se šalju podaci dobijeni sa akcelerometra i žiroskopa, formatirani na odgovarajući način. Na računaru, program pisan u *Python* prima podatke i skladišti ih u bazi podataka, ili ih daje *LSTM* modelu na evaluaciju, u zavisnosti od režima rada softvera.



Slika 2.5: Blok dijagram sistema

2.3 Akvizicija podataka

Akvizicija podataka se obavlja pritiskom na taster na razvojnoj ploči mikrokontrolera. Nakon prvog pritiska mikrokontroler putem serijske komunikacije kontinualno šalje podatke računaru, sve dok korisnik ponovo ne pritisne taster. Podaci se šalju u CSV formatu (comma separated values), sa periodom odabiranja od 5ms. Računar nakon primanja podataka može da ih sačuva u fajl koji predstavlja dataset ili da ih klasifikuje, u zavisnosti od toga u kom režimu radi.

Acc X	Acc Y	Acc Z	Gyro X	Gyro Y	Gyro Z
2464	61020	18572	6429	64634	65162
1036	63104	13652	5138	348	435
1532	64036	13840	3988	1701	1600
88	64812	14740	2766	1187	1991
64716	65100	15948	2062	80	2095
63448	412	18048	1163	65490	1870
62580	1744	20372	63761	65527	1441
62900	2476	21772	60120	65036	1234
64164	2864	19516	56954	64533	1380
60400	2576	20152	56434	63792	1358

Tablica 2.1: Primer početka sekvence cifre "7" koju MCU šalje na PC

Podaci su predstavljeni na 16 bita, a najduža sekvenca odbiraka je za cifru 8, i sastoji se od približno 200 odbiraka. Obzirom da se svaki odbirak sastoji od 6 brojeva, za najduži niz podataka imamo 19200 bita. Za prenos tih podatka od senzora do mikrokontrolera potrebno je približno 192 milisekunde (*I2C* je konfigurisan da ima *bitrate* od 100000 bita u sekundi), a za prenos od mikrokontrolera do računara približno 167 milisekundi (115200 *baud*). Ali obzirom da se podaci prosleđuju direkno računaru, bez akumulacije ili bilo kakve modifikacije vreme prenosa je oko 2 milisekunde (vreme prenosa jednog odbirka), zbog čega kažemo da sistem radi u realnom vremenu.

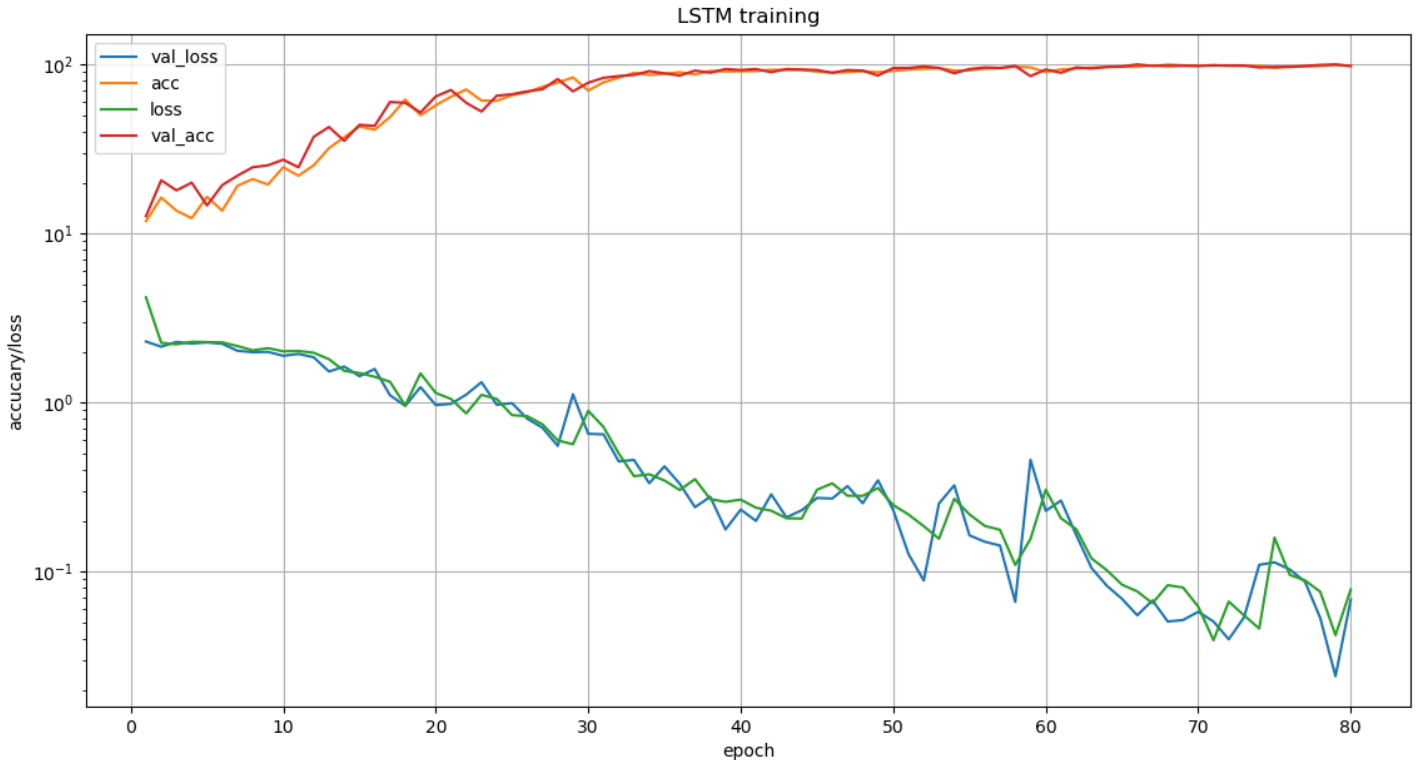
Dataset je formatiran u obliku matrice gde kolone predstavljaju svaki od 6 stepeni slobode (3 za akcelerometar, 3 za žiroskop), a redovi odbirke sekvence. Bilo je potrebno razlikovati i sekvence, zbog čega je uvedena kolona koja predstavlja broj vremenske sekvence (jedan primer napisanog broja). Pored toga, postoji i kolona u kojoj je zabeleženo o kom broju se radi. Te dve kolone se dodaju na strani računara, nakon prijema podataka. Dataset je pravljen broj po broj i za svaki broj postoji 60 primera, koji u proseku imaju po 100 odbiraka sa senzora.

Sample	Acc X	Acc Y	Acc Z	Gyro X	Gyro Y	Gyro Z	Digit
361	2464	61020	18572	6429	64634	65162	7
361	1036	63104	13652	5138	348	435	7
361	1532	64036	13840	3988	1701	1600	7
361	88	64812	14740	2766	1187	1991	7
361	64716	65100	15948	2062	80	2095	7
361	63448	412	18048	1163	65490	1870	7
361	62580	1744	20372	63761	65527	1441	7
361	62900	2476	21772	60120	65036	1234	7
361	64164	2864	19516	56954	64533	1380	7
361	60400	2576	20152	56434	63792	1358	7

Tablica 2.2: Primer sadržaja dataseta, za iste podatke kao u tabeli iznad

2.4 Obučavanje modela

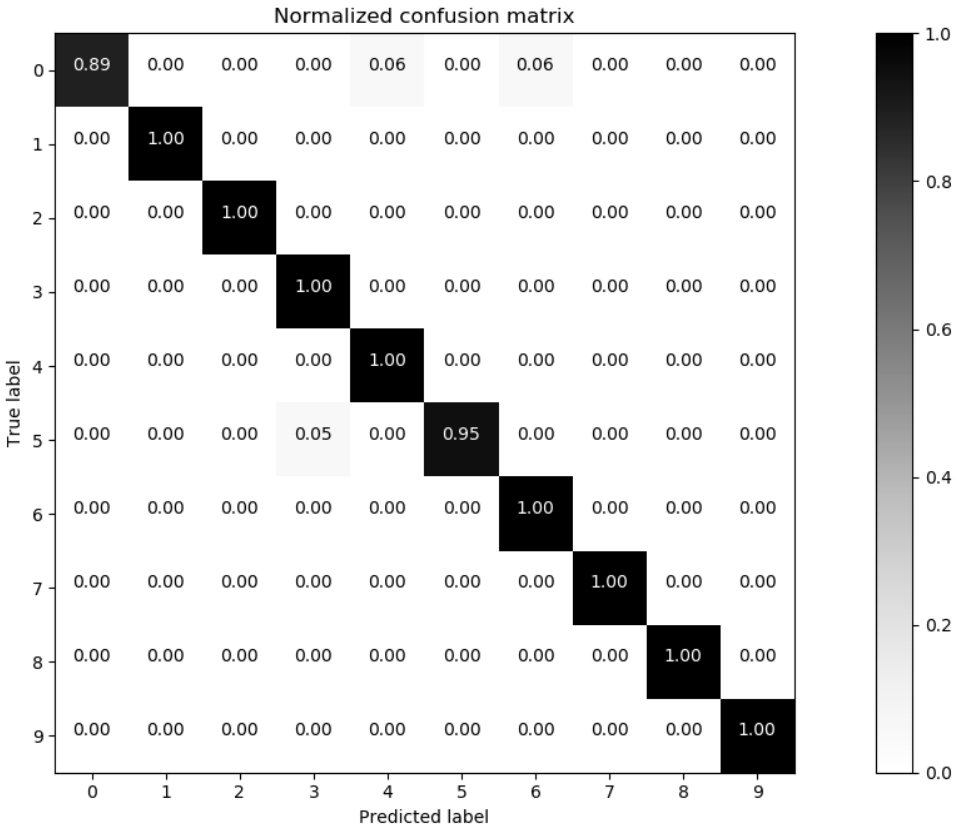
Za učitavanje dataseta korišćena je biblioteka *pandas*. Obzirom da sekvence nisu iste dužine i da *LSTM* zahteva da budu iste duzine, sve sekvence su dopunjene nulama do dužine najduže sekvence. Nakon toga, bilo je potrebno formatirati podatke u 3D niz, gde prva dimenzija prestavlja redni broj sekvence, druga redni broj odbirka u sekvenci, a treća o kom *feature*-u se radi. Obzirom da je klasifikacija višeklasna, labele je bilo potrebno predstaviti u *one-hot* kodovanju. Arhitektura modela sastoji se od 1300 neurona u skrivenom rekurentnom sloju i 10 *fully-connected* neurona u izlaznom sloju. Aktivaciona fukncija izlaznog sloja je *softmax*, a kriterijumska funkcija je *categorical cross-entropy*. Model se obučavao 80 epoha, pomoću *adam* optimizacionog algoritma, u grupama od po 32 odbirka. Za obučavanje su korišćene dve trećine dataseta, pri čemu je preostala trećina korišćena za validaciju.



Slika 2.6: Obučavanje modela

2.5 Rezultati obučavanja

Konfuziona matrica i F1 ocene za svaku klasu kao i ukupna F1 ocena dati su u nastavku. Ukupna F1 ocena računata je za sve klasifikacije zajedno, a ne kao srednja vrednost F1 ocena svake od klasa.



Slika 2.7: Konfuziona matrica

Cifra	0	1	2	3	4	5	6	7	8	9
F1	0.94	1.0	1.0	0.96	0.96	0.97	0.97	1.0	1.0	1.0

Tablica 2.3: F1 ocena za svaku od klasa

F1	0.98

Tablica 2.4: Ukupna F1 ocena

Zbog velikog broja neurona, model se dobro obučio na podacima, ali obzirom da je samo jedna osoba pisala cifre, na uniforman način, model ostvaruje dobre performanse samo kada se cifre pišu kao što su pisane pri kreiranju dataseta.

Obzirom da se svaka sekvenca dopunjuje nulama do dužine najduže sekvence dataseta, vreme klasifikacije je konstantno za svaku cifru koja se napiše, i iznosi približo 623 milisekunde, na poslovnom laptopu starom sedam godina, sa Intel Core i5-2520M procesorom, koji radi na 2.50GHz. Veličina modela je 78MB.

3. Zaključak i dalji razvoj

Testiranjem modela donet je zaključak da bi bilo korisno imati raznovrsnije podatke, čime bi se pokrili drugačiji načini pisanja cifara. Proširivanjem ideje, model ovakvog sistema bi se mogao preneti na pisanje cifara i durih alfanumeričkih karaktera, pravom olovkom, na kojoj bi se nalazili senzori kretanja. Bilo bi korisno da takav sistem poseduje bežičan prenos između olovke i računara. Stvar koja ostaje otvorena jeste primena takvog sistema.

Takođe, u trenutnoj implementaciji nije moguće koristiti cifre koje su pisane duže od najduže cifre dataseta. U slučaju da se napiše duža cifra, uzima se onoliko poslednjih odbiraka kolika je dužina najduže cifre dataseta, što nije solidno rešenje.

Bibliografija

[1] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

[2] S. Raschka, *Python Machine Learning*. Packt Publishing, 2015.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] W. McKinney, "pandas: a foundational python library for data analysis and statistics,"

[5] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[6] C. Olah, "Understanding lstm networks." http://colah.github.io/posts/2015-08-Understanding-LSTMs, 2015.