# IMPLEMENTACIJA JEDNOSTAVNE KONVOLUCIJSKE NEURONSKE MREŽE KORIŠTENJEM KERAS API-A

U ovom poglavlju će se implementirati jednostavne konvolucijske neuronske mreže korištenjem Keras API-a koji je podržan od strane TensorFlow 2 platforme za strojno učenje. Prvo će se ukratko opisati Keras API. Nakon toga će se navesti i opisati aktivacijske funkcije u situacijama kada se one zovu kao argument već postojećeg sloja i kada se one zovu kao zasebni, aktivacijski, slojevi.

Sam praktični dio će se baviti usporedbom jednostavnih konvolucijskih mreža kada se koriste različiti aktivacijski slojevi s različito postavljenim parametrima.

## KERAS API

Keras je API za duboko učenje napisan u Python programskom jeziku. Pokreže se na TensorFlow platformi za strojno učenje. Ovaj API je razvijen kako bi inženjerima i znanstvenicima omogućio brzo eksperimentiranje tijekom istraživanja.

### KERAS I TENSORFLOW 2

TensorFlow 2 je *end-to-end*, *open-source* platforma za strojno učenje. Ona kombinira 4 ključne sposobnosti:

1. Efikasno izvodi jednostavne operacije s matricama korištenjem CPU-a, GPU-a ili TPU-a.
2. Izračunava gradijent proizvoljnih izraza koji imaju sposobnost deriviranja.
3. Skalira izračune na mnogim uređajima.
4. Izvozi programe (grafove) na vanjske *runtime*-ove poput servera, web preglednika, mobilnih i ugradbenih uređaja.

Keras je TensorFlow-ov API visoke razine. To je jako pristupačno, visoko-produktivno sučelje koje se koristi za rješavanje problema strojnog učenja i koje je specijalizirano za duboko učenje. Pruža bitne apstrakcije i gradivne blokove za razvoj i izvoz rješenja strojnog učenja s velikom brzinom izvođenja iteracija.

Keras se može izvoditi na TPU-ima (*Tensor Processing Unit*) i velikim grozdovima GPU-a (*Graphic Processing Unit*). On dolazi u paketu s TensorFlow 2 platformom te mu se pristupa korištenjem izraza tensorflow.keras. Za korištenje Keras API-a, potrebno je samo instalirati TensorFLow 2.

Keras/TensorFlow 2 je kompatibilan s:

1. Python-ovim verzijama 3.5-3.8
2. Ubuntu 16.04 i kasnijim verzijama
3. Windows 7 i kasnijim verzijama
4. macOs 10.12.6 (Sierra) i kasnijim verzijama

Keras (κέρας) na grčki znači 'rog'. Njegovo ime je referenca na pjesničku sliku iz antičke grčke i latinske književnosti koja se prvi put susreće u Homerovoj Odiseji. Mitološki duhovi (*Oneiros*) koji prolaze kroz vrata roga objavljuju budućnost koja će se sigurno dogoditi. Keras je inicijalno bio razvijen u sklopu istraživanja na projektu ONEIROS (*Open-ended Neuro-Electronic Intelligent Robot Operating System*).[1]

## AKTIVACIJSKE FUNKCIJE U KERAS API-U

Aktivacijske funkcije se mogu koristiti odvojeno u aktivacijskom sloju ili korištenjem 'activation' argumenta u ostalim slojevima u neuronskoj mreži.

*Placeholder za primjer*

Ovo je jednako sljedećem kodu:

*Placeholder za inačicu*

Sve aktivacijske funkcije mogu se pozvati korištenjem njihovog *string* identifikatora:

*Placeholder za string identifikator*[2]

### RELU AKTIVACIJSKA FUNKCIJA

*Placehoder za kod s relu*

Argumenti ReLU aktivacijske funkcije su:

1. x: ulazna matrica ili varijabla
2. alpha: decimalni broj koji upravlja nagibom za vrijednosti koje su niže od vrijednosti praga
3. max\_value: decimalni broj kojim se postavlja najveća vrijednost koju će funkcija vratiti
4. threshold: decimalni broj kojim se postavlja vrijednost praga ispod koje će aktivacijska funkcija vratiti nulu

Ova funkcija na izlazu vraća matricu jednakih dimenzija kao i ulazna matrica x. Ta matrica predstavlja ulaznu matricu čije su vrijednosti izmijenjene korištenjem ReLU aktivacijske funkcije.[2]

### SIGMOIDNA AKTIVACIJSKA FUNKCIJA

*Placeholder za sigmoid*

Za male vrijednosti (manje od pet), sigmoidna funkcija će vratiti vrijednosti koje se nalaze blizu nule. Za velike vrijednosti (veće od pet) funkcija će vratiti vrijednosti koje se nalaze blizu jedan.

Sigmoidna aktivacijska funkcija ekvivalentna je softmax funkciji koja prima 2 argumenta, ali gdje je drugi argument postavljen na vrijednost nula. Sigmoidna funkcija će uvijek vratiti vrijednost koja se nalazi između nula i jedan.

Argumenti sigmoidne aktivacijske funkcije su:

1. x: ulazna matrica ili varijabla

Ova funkcija vraća matricu jednakih dimenzija kao i ulazna matrica. Vrijednosti izlazne matrice rezultat su djelovanja sigmoidne funkcije na vrijednosti ulazne matrice.[2]

### SOFTMAX AKTIVACIJSKA FUNKCIJA

*Placeholder za softmax*

Softmax funkcija uzima vektor realnih vrijednosti i vraća vektor kategoričkih vjerojatnosti. Elementi izlaznog vektora se nalaze u opsegu (0, 1). Zbroj svih elemenata izlaznog vektora daju jedan. Ako funkcija prima više vektora, ona će sa svakim vektorom raditi pojedinačno.

Softmax funkcija se često koristi kao aktivacijska funkcija u posljednjem sloju u neuronskoj mreži jer se koristi za klasifikaciju ulaznih primjera. Razlog je što se njen rezultat može tumačiti kao distribucija vjerojatnosti.

Argumenti softmax aktivacijske funkcije su:

1. x: ulazna matrica ili varijabla
2. axis: cjelobrojna vrijednost, označava os uz koju se primjenjuje softmax funkcija

Ova funkcija na izlazu vraća matricu čije su vrijednosti ne-negativne i čiji cjelokupni zbroj daje jedan.

Ovdje je potrebno napomenuti da korištenje ove funkcije može uzrokovati ValueError grešku ako je dimenzija matrice dim(x)==1.[2]

### SOFTPLUS AKTIVACIJSKA FUNKCIJA

*Placeholder za softplus*

Softplus aktivacijska funkcija izražava se kao: .

Argumenti softplus aktivacijske funkcije su:

1. x: ulazna matrica

Ova funkcija vraća softplus aktivaciju.

### SOFTSIGN AKTIVACIJSKA FUNKCIJA

*Placeholder za softsign*

Softsign aktivacijska funkcija izražava se kao: .

Argumenti softsign aktivacijske funkcije su:

1. x: ulazna matrica

Ova funkcija na izlazu daje vrijednost softsign aktivacije.

### TANH AKTIVACIJSKA FUNKCIJA

*Placeholder za tanh*

Argumenti tanh (tangens hiperboli) funkcije su:

1. x: ulazna matrica

Ova funkcija na izlazu daje matricu jednakog oblika i s istim tipovima vrijednosti kao i ulazna matrica x.[2]

### SELU AKTIVACIJSKA FUNKCIJA

*Placeholder za selu*

SELU je kratica od *Scaled Exponential Linear Unit* i definirana je kao:

* ako je x>0: vrati
* ako je x<0: vrati

Gdje su alpha i scale predefinirane konstante (alpha=1.67326324, scale=1.05070098).

SELU aktivacijska funkcija množi konstantu scale s izlazom tf.keras.activations.elu aktivacijske funkcija kako bi se dobio veći nagib za pozitivne ulaze.

Vrijednosti alpha i scale odabrane su na način da srednje vrijednosti i varijance ulaza budu sačuvane između dva uzastopna sloja dokle god su težine veza ispravno inicijalizirane i dokle god je broj ulaza 'dovoljno velik'.

Argumenti SELU funkcije su:

1. x: ulazna matrica ili skalarna vrijednost

Funkcija na izlazu daje skaliranu ELU aktivaciju: scale\*elu(x, alpha).[2]

### ELU AKTIVACIJSKA FUNKCIJA

*Placeholder za elu*

ELU je kratica od *Exponential Linear Unit*.

Ako je parametar alpha veći od nule, funkcija vraća sljedeće vrijednosti:

1. x ako je x>0
2. alpha\*(exp(x)-1) ako je x<0

ELU hiper-parametar alpha kontrolira vrijednost za koju će se ELU zasititi kod negativnih ulaza u mrežu. ELU funkcija rješava problem nestajućeg gradijenta.

ELU funkcija će vratiti negativne vrijednosti koje će srednju vrijednost aktivacije pomicati prema nuli. Srednje vrijednosti aktivacija koje se nalaze bliže nuli omogućuju brže učenje jer gradijent pomiču bliže prirodnom gradijentu.

Argumenti ELU funkcije su:

1. x: ulazna matrica
2. alpha: skalar, nagib kod negativne sekcije

Ova funkcija na izlazu daje vrijednost ELU funkcije.[2]

### EXPONENTIAL AKTIVACIJSKA FUNKCIJA

*Placeholder za exponential*

Argumenti koje prima exponential funkcija su:

1. x: ulazna matrica

Ova funkcija na izlazu daje matricu čije su vrijednosti rezultat djelovanja eksponencijalne funkcije exp(x) na vrijednosti ulazne matrice.[2]

## AKTIVACIJSKI SLOJEVI U KERAS API-U

Aktivacijski slojevi Keras API-a većinom čine ReLU i njene inačice s izuzetkom softmax i ELU sloja.

### ReLU SLOJ

*Placeholder za ReLU sloj*

S osnovno postavljenim vrijednostima, ovaj sloja vraća max(x, 0), dok inače slijedi pravila:

*Placeholder za pravila*

S tim da je input\_shape proizvoljan argument koji se koristi kada je ReLU sloj prvi skriveni sloj u neuronskoj mreži.

Izlaz iz ReLU sloja istog je oblika kao i ulaz.

Argumenti ReLU sloja su:

1. max\_value: decimalni broj koji je veći ili jednak nuli; predstavlja maksimalnu izlaznu vrijednost iz sloja. Osnovna postavka ovog parametra je postavljena na None što znači da je ona neograničena.
2. negative\_slope: decimalni broj koji je veći ili jednak nuli; predstavlja koeficijent negiba za negativne brojeve. Osnovna postavka ovog parametra je postavljena na nulu što znači da negativne vrijednosti neće imati nagib.
3. threshold: decimalni broj; predstavlja vrijednost praga ispod kojeg će neuron postati neaktivan. Osnovna postavka ovog parametra je postavljena na nulu. [3]

### SOFTMAX SLOJ

*Placeholder za softmax*

S tim da je input\_shape proizvoljan argument koji se koristi kada je softmax sloj prvi skriveni sloj u neuronskoj mreži. Izlaz ovog sloja jednakog je oblika kao i ulaz u sloj.

Argumenti softmax sloja su:

1. axis: cijeli broj ili lista cjelobrojnih vrijednosti. To je os duž koje se primjenjuje softmax normalizacija

Ostali, proizvoljni, argumenti su:

1. inputs: ulazi u softmax sloj
2. mask: maska s bool vrijednostima; istog je oblika kao i inputs argumenti. Osnovna vrijednost je postavljena na None. [3]

### LeakyReLU SLOJ

*Placeholder za leakyrelu*

S tim da je input\_shape proizvoljan argument koji se koristi kada je LeakyReLU sloj prvi skriveni sloj u neuronskoj mreži. Izlaz ovog sloja jednakog je oblika kao i ulaz u sloj.

LeakyReLU sloj ostvaruje mali gradijent kada neuron nije aktiviran čime se nastoji izbjeći problem nestajućeg ReLU-a.

*Placeholder za situacije*

Argumenti LeakyReLU sloja su:

1. alpha: decimalni broj koji je veći ili jednak od nule; predstavlja koeficijent nagiba za negativne vrijednosti. Osnovna vrijednost mu je postavljena na 0.3. [3]

### PReLU SLOJ

*Placeholder za prelu*

S tim da je input\_shape proizvoljan argument koji se koristi kada je PReLU sloj prvi skriveni sloj u neuronskoj mreži. Izlaz ovog sloja jednakog je oblika kao i ulaz u sloj.

PReLU je parametarska verzija ReLU aktivacije koja slijedi pravila:

*Placeholder za pravila*

Gdje je alpha parametar kojeg mreža uči tijekom procesa učenja i koji je istog oblika kao i ulaz u sloj x.

Argumenti PReLU sloja su:

1. alpha\_initializer: inicijalizacijska funkcija koja služi za postavljanje inicijalnih vrijednosti težina veza
2. alpha\_regularizer: regulator težina
3. alpha\_constant: ograničenja težina
4. shared\_axes: osi uz koje se dijele naučeni parametri aktivacijske funkcije. Na primjer, ulazna matrica značajki iz 2D konvolucije ima oblik (batch, height, width, channels) i cilj je da se ti parametri prostorno podijele tako da svaki filter ima samo jedan skup parametara. Tada se ovaj argument postavi na vrijednost shared\_axes=[1, 2]. [3]

### ELU SLOJ

*Placeholder za elu*

S tim da je input\_shape proizvoljan argument koji se koristi kada je PReLU sloj prvi skriveni sloj u neuronskoj mreži. Izlaz ovog sloja jednakog je oblika kao i ulaz u sloj.

Ova funkcija slijedi pravila:

*Placeholder za pravila elu-a*

Argumenti ELU sloja su:

1. alpha: koeficijent za negativni faktor[3]

### ThresholdedReLU SLOJ

*Placeholder za trelu*

S tim da je input\_shape proizvoljan argument koji se koristi kada je PReLU sloj prvi skriveni sloj u neuronskoj mreži. Izlaz ovog sloja jednakog je oblika kao i ulaz u sloj.

Ova funkcija slijedi pravila:

*Placeholder za pravila trelua*

Argumenti ThresholdedReLU sloja su:

1. theta: decimalni broj koji je veći ili jednak nuli; predstavlja lokaciju praga aktivacije.[3]

## LITERATURA

[1] <https://keras.io/about/>

[2] <https://keras.io/api/layers/activations/>

[3] <https://keras.io/api/layers/activation_layers/>