Primjena neuronskih mreža na kvalifikaciju slika pomoću MATLAB-a

Majstorović, Petra

SVEUČILIŠTE U DUBROVNIKU
ODJEL ZA ELEKTROTEHNIKU I RAČUNARSTVO

Petra Majstorović

Primjena neuronskih mreža na klasifikaciju slika pomoću MATLAB-a

Classification of images by neural networks in MATLAB

ZAVRŠNI RAD

Dubrovnik, 2023.

SVEUČILIŠTE U DUBROVNIKU
ODJEL ZA ELEKTROTEHNIKU I RAČUNARSTVO

Primjena neuronskih mreža na klasifikaciju slika pomoću MATLAB-a

Classification of images by neural networks in MATLAB

ZAVRŠNI RAD

Kolegij: Matematički programski alati, Vjerojatnost i statistika
Studij: Elektrotehnika i računarstvo
Vrsta studija: sveučilišni
Razina: preddiplomski
Studijski smjer: Primijenjeno/poslovno računarstvo
Mentor: prof. dr. sc. Martin Lazar
Student: Petra Majstorović
JMBAG: 0177057002

Dubrovnik, Rujan 2023.

SAŽETAK

S rastom njihove popularnosti duboko učenje i neuronske mreže se sve više koriste u svakodnevnom životu. U ovom radu cilj je objasniti što su točno neuronske mreže, na koji način rade te objasniti primjenu obučene neuronske mreže *GoogLeNet* u MATLAB-u na primjeru klasifikacije slika. Objasnit ćemo pojam prijenosa učenja te kako se koristi na danom primjeru. Preko rezultata obuke pokazat ćemo da je ovaj način učenja neuronske mreže uspješan u učenju na malom broju ulaznih podataka.

Ključne riječi: MATLAB, klasifikacija slika, neuronska mreža, duboko učenje, GoogLeNet

ABSTRACT

With the rise of the popularity of deep learning and neural network their use in everyday life is expanding. Goal of this paper is to explain what neural networks are, how they work and explain their use with the example of image classification in MATLAB using pretrained *GoogLeNet* neural network. Additionally, we explain what is transfer learning and how it is used on a given example. Examining the results of training we will show that this way of training a neural network is successful in learning given a small amount of input data.

Key words: MATLAB, classification of images, neural network, deep learning, GoogLeNet

SADRŽAJ:

1. UVOD

S porastom količine informacija u svakodnevnom životu sve ih je teže kvalitetno obraditi i iskoristiti. Iz tog razloga raste potreba za inteligentnim sustavom koji bi te informacije mogao analizirati i obraditi. Najpopularniji pristupi obradi podataka su rješenja dubokog učenja iz područja umjetne inteligencije. Duboko učenje se sve više koristi u aspektima svakodnevnog života. Od Internet pretraživača do predviđanja cijena učestalo se koristi u gotovo svakom zanimanju. Stoga je važno osnovno teorijsko znanje i razumijevanje dubokog učenja. Duboko učenje na temelju ulaznih podataka koristeći model neuronskih mreža predviđa ishode te nam tako olakšava svakodnevni život.

S obzirom da je duboko učenje specifična podvrsta strojnog učenja na početku ćemo objasniti što je strojno učenje. U nastavku drugog poglavlja ćemo objasniti na čemu su temeljene neuronske mreže, opisati od kojih dijelova se sastoji njihov model te navesti najpoznatije aktivacijske funkcije. Na kraju poglavlja ćemo iznijeti razlike u izradi modela neuronske mreže te napraviti kratki osvrt na strojno učenje, duboko učenje i neuronske mreže.

U trećem poglavlju ćemo na primjeru objasniti proces izrade neuronske mreže koristeći MATLAB razvojno okruženje. U našem primjeru koristit ćemo *Deep Networks Designer* za klasifikaciju slika.

2. Neuronske mreže

U ovom poglavlju objasnit ćemo što su neuronske mreže, na čemu su utemeljene te kako rade. Također ćemo objasniti njihovu povezanost s dubokim učenjem.

2.1. Strojno učenje

Strojno učenje je podvrsta umjetne inteligencije i računalne znanosti koja koristi podatke i algoritme kako bi imitirala način na koji ljudi uče. S vremenom povećava svoju točnost te omogućuje softverskim aplikacijama bolje predviđanje ishoda bez prethodnog programiranja [1]. Algoritmi strojnog učenja koriste prošle podatke kao ulazne vrijednosti kako bi predvidjeli nove izlazne vrijednosti.

Često se koristi u mehanizmima za preporuke, kao što su Netflix i YouTube te u prepoznavanju prijevara, filtriranju neželjene pošte, otkrivanju zlonamjernih softverskih prijetnji i automatizaciji uredskog poslovanja.

Važno je jer daje poduzetnicima uvid u trendove o ponašanju potrošača i obrasce poslovanja te za razvoj novih proizvoda. Velik broj današnjih vodećih kompanija čine strojno učenje centralnim dijelom svog poslovanja što ga čini značajnim konkurentskim razlikovnim čimbenikom za mnoge kompanije.

Dijeli se na kategorije u ovisnosti na koji način algoritam uči kako biti što precizniji u svojim predviđanjima. Četiri glavna pristupa su nadzirano učenje, nenadzirano učenje, djelomično nadzirano učenje i pojačano učenje.

2.1.1. Nadzirano učenje

Pri korištenju ovog tipa strojnog učenja algoritmu se daju označeni podatci za učenje i definirane varijable kojima algoritam treba odrediti korelaciju. U ovom načinu učenja i ulazne i izlazne varijable su unaprijed poznate.

2.1.2. Nenadzirano učenje

Ovaj tip strojnog učenja sastoji se od učenja na neoznačenim podatcima. Algoritam skenira podatke kako bi našao značajnu korelaciju. U ovom pristupu zadatak je otkriti strukturu podataka. Nenadziranim učenjem može se smatrati i vizualizacija podataka.

2.1.3. Djelomično nadzirano učenje

U ovom pristupu koristi se spoj dva prethodna tipa strojnog učenja. Algoritmu se daju većinom označeni podatci koji služe kao vodič za klasifikaciju i otkrivanje korelacije. Na ovaj način se rješava problem malog broja označenih podataka. Model može slobodno istraživati podatke te samostalno razviti vlastito razumijevanje.

2.1.4. Pojačano učenje

Pojačano učenje je slično modelu nadziranog učenja, ali se algoritam ne uči na uzorku podataka. Model uči na principu pokušaja i pogrešaka (engl. *trial and error*). Koristi se kako bi naučili stroj da obavlja sekvencijalne procese za koje postoje predodređena pravila. Algoritam se programira kako bi izvršio zadatak i dao pozitivne ili negativne znakove dok radi. Pozitivan ishod se „ojačava“ kako bi se razvila preporuka za daljnje učenje. Većinom algoritam sam odlučuje kojim koracima ići na putu do rješenja.

2.2. Duboko učenje

Duboko učenje je metoda strojnog učenja u kojem računala uče na temelju primjera. Računalni model obavlja izravnu klasifikaciju iz slika, teksta ili zvuka uz pomoć dubokog učenja. Mogu postići najsuvremeniju točnost koja nekad premašuje performanse ljudi. Modeli se obučavaju korištenjem velikog skupa označenih podataka i koristeći višeslojnu arhitekturu neuronskih mreža [2].

Duboko učenje je podskup strojnog učenja koji koristi algoritme umjetne neuronske mreže, koje su inspirirane strukturom i funkcijom mozga i sposobni su za samoučenje. Umjetne neuronske mreže su trenirane za „učenje“ modela i obrazaca bez da im se eksplicitno reče kako riješiti problem [3].

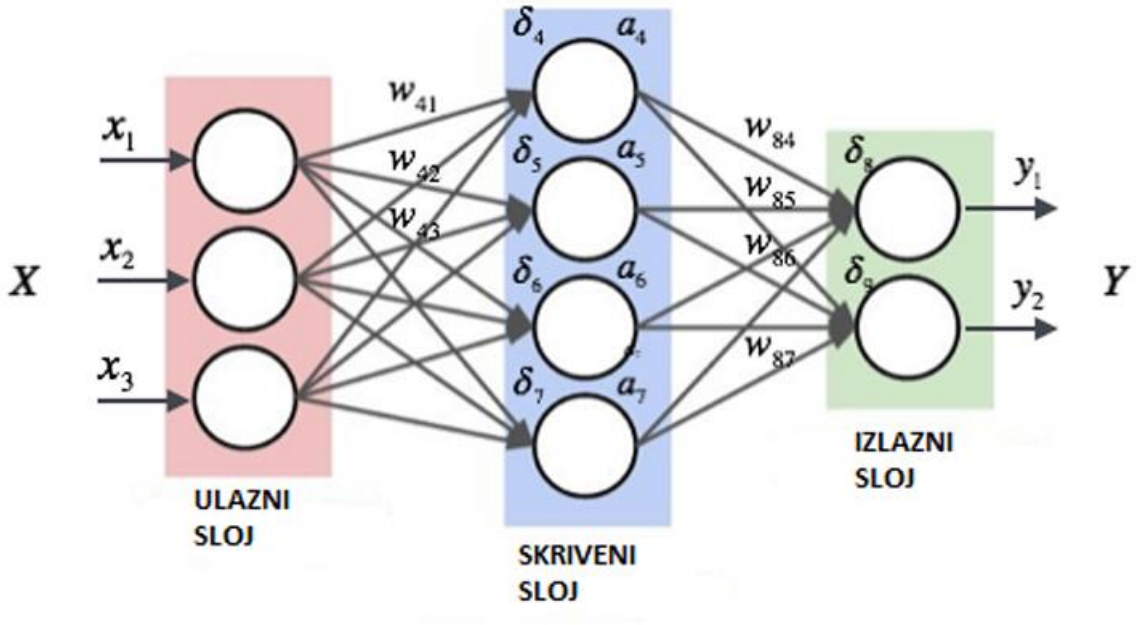
2.3. Što su neuronske mreže?

Neuronske mreže ili umjetne neuronske mreže su podskup strojnog učenja te se nalaze u središtu algoritama dubokog učenja. Njihovo ime i struktura inspirirana je ljudskim mozgom te oponaša kako biološki neuroni signaliziraju jedan drugome.

Glavni elementi neuronske mreže su:

- Ulazni sloj (engl. *input layer*) je sloj koji uzima neobrađeni unos iz domene. Na ovom sloju se ne izvode proračuni. Čvorovi ovdje samo prosljeđuju informacije, tj. značajke skrivenom sloju.
- Skriveni sloj (engl. *hidden layer*) je sloj čiji čvorovi nisu izloženi. Oni pružaju apstrakciju neuronskoj mreži. Skriveni sloj izvodi sve vrste izračuna na značajkama unesenim kroz ulazni sloj i prenosi rezultat na izlazni sloj.
- Izlazni sloj (engl. *output layer*) je posljednji sloj mreže koji donosi informacije naučene kroz skriveni sloj i daje konačnu vrijednost kao rezultat.

Primjer strukture neuronske mreže možemo vidjeti na slici ispod.



Slika 1. Struktura neuronske mreže

Kada učimo o neuronskim mrežama, naići ćemo na dva bitna pojma koji opisuju kretanje informacija *feedforward* i *backpropagation*, tj. propagacija naprijed i širenje unazad.

U propagaciji naprijed (engl. *feedforward propagation*) protok informacija odvija se u smjeru prema naprijed. Ulaz se koristi za izračun neke srednje funkcije u skrivenom sloju koja se zatim koristi za izračun izlaza. Kod širenja unatrag (engl. *backpropagation*) težine mrežnih veza se konstantno prilagođavaju kako bi se smanjila razlika između stvarnog izlaznog vektora mreže i željenog izlaznog vektora.

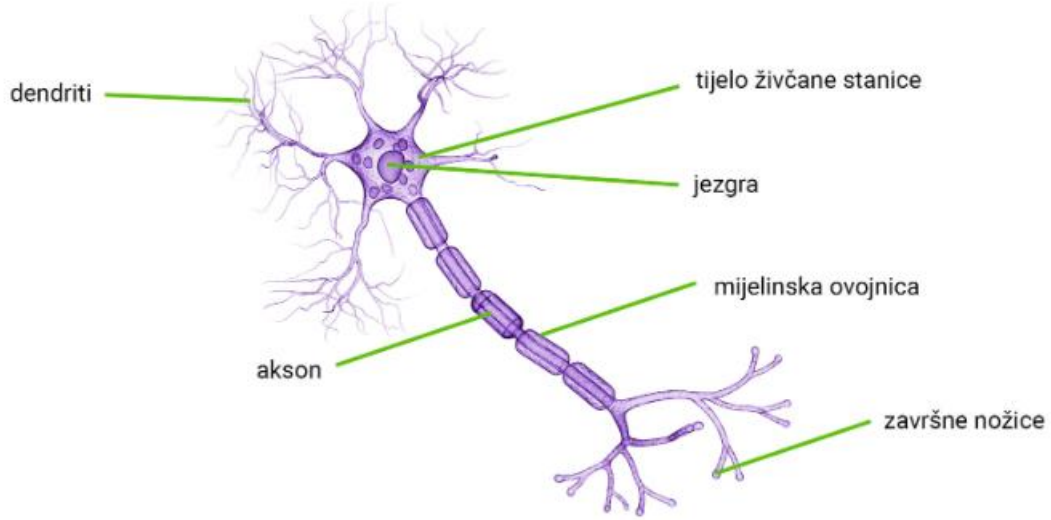
Kako bi učile i poboljšale svoju točnost tijekom vremena neuronske mreže oslanjaju se na podatke za obuku. Oni su učinkoviti alati u računalnoj znanosti i umjetnoj inteligenciji za brzu klasifikaciju i grupiranje podataka. U usporedbi s ručnom identifikacijom od strane ljudskih

stručnjaka, zadaci prepoznavanja govora ili slike mogu se izvršiti za nekoliko minuta umjesto sati. Algoritam Google pretraživanja jedna je od najpoznatijih neuronskih mreža [1].

2.4. Umjetni neuron

S rastom popularnosti umjetne inteligencije počela se voditi rasprava o načinu na koji natjerati računalo da „uči“. Jedan od pristupa je bio da računala oponašaju način na koji ljudi uče. Budući da je mozak primarno skup međusobno povezanih živčanih stanica, pojavila se ideja o povezivanju jednostavnih umjetnih živčanih stanica na složene načine kako bi se postigli složeni rezultati.

Ljudski mozak ima milijarde živčanih stanica, tj. neurona. Neuroni su međusobno povezane živčane stanice u ljudskom mozgu koje obrađuju i prenose kemijske i električne signale. Sastoje se od dendrita, stanične jezgre, aksona i sinapsa. Dendriti su grane koje primaju informacije od drugih neurona. Stanična jezgra obrađuje informacije primljene od dendrita, a akson je produžetak neurona koji se koristi za slanje informacija. Sinapsa je veza između aksona i drugih neuronskih dendrita. Izgled neurona prikazan je na slici 2.



Slika 2. Neuron [4]

Istraživači Warren McCullock i Walter Pitts objavili su prvi koncept pojednostavnjene živčane stanice u 1943. godini. Koncept je nazvan McCullock-Pitts (krat. MCP) neuron.

Taj neuron opisan je kao jednostavna logička vrata s binarnim izlazima. Više signala dolazi do dendrita i integriraju se u jezgru stanice te ako akumulirani signal prijeđe određeni prag generira se izlazni signal koji se dalje prosljeđuje preko aksona [5].

Umjetni neuron je matematička funkcija bazirana na modelu biološkog neurona gdje svaki neuron prima ulazne podatke, zasebno ih promatra, zbraja te prosljeđuje zbroj aktivacijskoj funkciji kako bi dobio izlazni podatak.

Umjetni neuron ima slijedeće karakteristike:

- Neuron je matematička funkcija modelirana prema živčanoj stanici
- On je elementarna jedinica u umjetnoj neuronskoj mreži
- Jedan ili više ulaznih podataka se promatraju odvojeno
- Ulazni podaci se zbrajaju i prolaze kroz aktivacijsku funkciju kako bi se dobio izlazni podatak
- Svaki neuron ima aktivacijski signal
- Svaka poveznica prenosi informaciju o ulaznom podatku
- Svaki neuron je povezan s drugim neuronom putem poveznice [5].

2.5. Perceptron

Perceptron je naziv za jednostavni model neuronske mreže s aktivacijskom funkcijom praga o kojoj ćemo govoriti kasnije. Bio je to jedan od prvih službenih modela izračuna neurona, a s obzirom na njegovu ključnu ulogu u povijesti neuronskih mreža zove se i „majkom svih umjetnih neuronskih mreža".

Perceptron je temeljni koncept strojnog učenja te se smatra jednom od najboljih i najspecifičnijih vrsta umjetnih neuronskih mreža. Osnovne komponente perceptrona su:

1. Ulazni sloj: Ulazni sloj se sastoji od jednog ili više ulaznih neurona koji primaju ulazne signale iz vanjskog svijeta ili drugih slojeva neuronske mreže.
2. Težine: Svaki ulazni neuron povezan je s težinom koja predstavlja snagu veze između ulaznog i izlaznog neurona.
3. Statistička pristranost: Ona se dodaje ulaznom sloju kako bi se perceptronu pružila dodatna fleksibilnost u modeliranju složenih uzoraka u ulaznim podacima.
4. Aktivacijska funkcija: Aktivacijska funkcija određuje izlaz perceptrona na temelju izračunanog zbroja ulaza i pristranog člana. Uobičajene aktivacijske funkcije koje se koriste u perceptronima su binarna step funkcija, sigmoidna funkcija i ReLU funkcija o kojima ćemo više govoriti kasnije.
5. Izlaz: Izlaz perceptrona je binarna vrijednost, 0 ili 1, koja označava klasu ili kategoriju kojoj ulazni podaci pripadaju.
6. Algoritam obuke: Perceptron se obično trenira korištenjem algoritma nadziranog učenja kao što je algoritam učenja perceptrona ili širenje unatrag. Tijekom učenja

težine i pristranosti perceptrona se prilagođavaju kako bi se smanjila pogreška između predviđenog izlaza i stvarnog izlaza za dani skup primjera učenja.

Perceptron je najjednostavniji oblik umjetne neuronske mreže i najčešće se koristi za prepoznavanje govora i klasifikaciju slika. Moćan je algoritam koji je otvorio put složenijim neuronskim mrežama koje se danas koriste u dubokom učenju [5].

Izgled perceptrona prikazan formulom:

$$y = f\left(\sum w_i\ x_i + b_i\right)$$

gdje je f aktivacijska funkcija, x_n ulaz neurona, w_n težina, a b_n statistička pristranost.

2.6. Izrada modela

Osnovni model umjetnog neurona sastoji se od niza prilagodljivih parametara koji se zovu težinske vrijednosti, tj. težine. Te se težinske vrijednosti upotrebljavaju kao multiplikatori za ulazne varijable neurona koje se zbrajaju. Zbroj težina pomnožen s ulaznim varijablama pristranosti naziva se linearna kombinacija ulaznih varijabli.

Ako neuron koji promatramo ima šest ulaza, ulaz1 do ulaz6, treba nam i šest težina. Nazvat ćemo ih težina1 do težina6. Osim toga obično treba uvesti i pojam statističke pristranosti. Linearnu kombinaciju zatim možemo izračunati na sljedeći način:

linearna kombinacija = statistička pristranost + težina1 × ulaz1 + ... + težina6 × ulaz6 [6].

2.7. Aktivacijske funkcije

Aktivacijska funkcija je funkcija koja se koristi za dobivanje izlaza čvora. Također je poznata kao prijenosna funkcija. Svrha aktivacijske funkcije je dodati nelinearnost neuronskoj mreži. Aktivacijske funkcije uvode dodatni korak na svakom sloju neuronske mreže.

Pretpostavimo da imamo neuronsku mrežu koja radi bez aktivacijskih funkcija. U tom će slučaju svaki neuron izvoditi samo linearnu transformaciju na ulazima koristeći težine i statističke pristranosti. To je zato što nije važno koliko skrivenih slojeva povezujemo u neuronsku mrežu; svi slojevi će se ponašati na isti način jer je kompozicija dviju linearnih funkcija sama po sebi linearna funkcija.

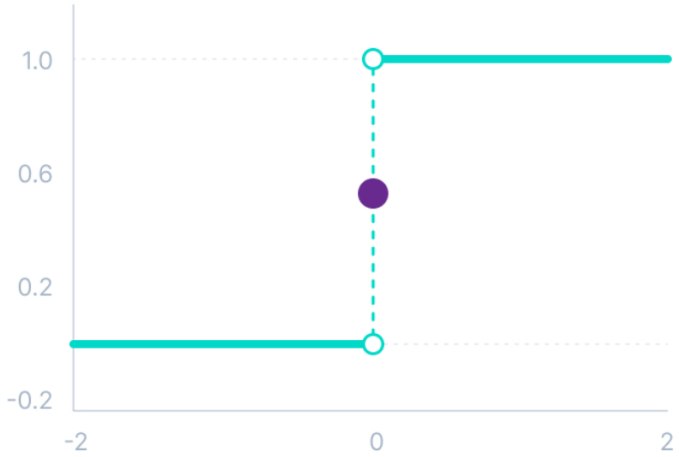
Iako neuronska mreža postaje jednostavnija, učenje bilo kojeg složenog zadatka je nemoguće, a naš bi model bio samo model linearne regresije [7].

Aktivacijske funkcije možemo podijeliti u tri skupine:

1. Binarna step funkcija (engl. *Binary Step Function*)
2. Linearna aktivacijska funkcija
3. Nelinearne aktivacijske funkcije

Binarna step funkcija ovisi o vrijednosti praga koja odlučuje hoće li se neuron aktivirati ili ne. Unos doveden u funkciju aktivacije uspoređuje se s određenim pragom; ako je ulaz veći od njega, tada je neuron aktiviran, inače je deaktiviran, što znači da se njegov izlaz ne prosljeđuje na sljedeći skriveni sloj.

Binarna step funkcija ne može pružiti izlaze s više vrijednosti, na primjer, ne može se koristiti za probleme klasifikacije s više klasa. Na slikama ispod možemo vidjeti grafički izgled binarne step funkcije kada je vrijednost praga 0.



Slika 3. Binarna step funkcija [7]

Izgled binarne step funkcije prikazan je sljedećom formulom:

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

Linearna funkcija je oblika:

$$f(x) = cx$$

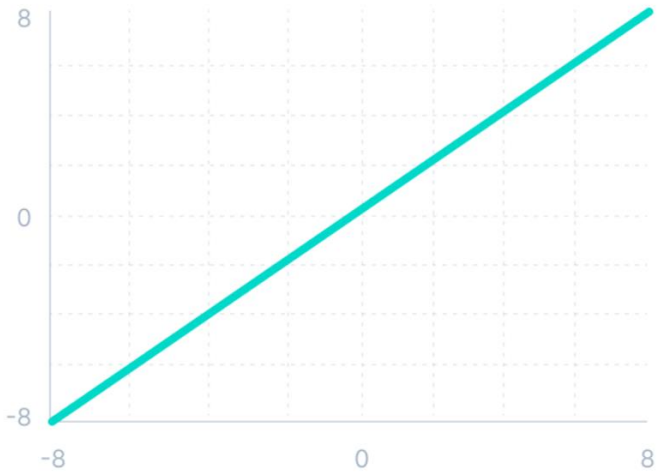
gdje je c skalarna vrijednost.

Linearna aktivacijska funkcija također poznata kao "funkcija bez aktivacije" (ili "funkcija identitete" u slučaju kada je c=1) je funkcija gdje je aktivacija proporcionalna ulazu. Funkcija ne radi ništa s dobivenim zbrojem ulaza već jednostavno vrati vrijednost koju je dobila.

U dubokom učenju linearna aktivacijska funkcija se koristi kada želimo da izlazna vrijednost bude jednaka ulazu. U većini slučajeva nije prikladno koristiti ovu aktivacijsku funkciju.

Ukoliko se koristi u svakom sloju neuronske mreže istu će pretvoriti u jedan sloj. Korisna je u slučajevima kada je to potrebno ili kada se koriste druge aktivacijske funkcije u skrivenim slojevima neuronske mreže.

Ispod prikazana linearna aktivacijska funkcija jednostavno je model linearne regresije. Zbog svojih ograničenih sposobnosti, ona ne dopušta modelu stvaranje složenih preslikavanja između ulaza i izlaza mreže.



Slika 4. Linearna aktivacijska funkcija [7]

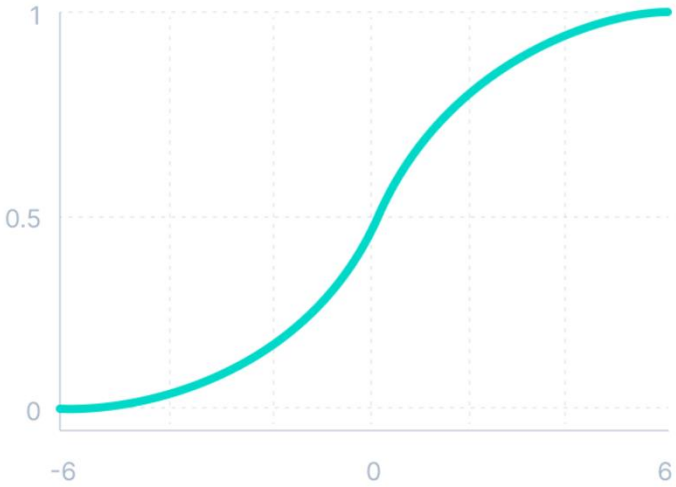
Nelinearne aktivacijske funkcije uklanjaju ograničenja linearnih aktivacijskih funkcija.

Nelinearne aktivacijske funkcije dopuštaju širenje unazad te omogućuju slaganje više slojeva neurona jer bi izlaz sada bio nelinearna kombinacija ulaza koji prolazi kroz više slojeva. Bilo koji izlaz može se predstaviti kao funkcionalno izračunavanje u neuronskoj mreži [7]. Dalje ćemo navesti nekoliko primjera nelinearnih aktivacijskih funkcija.

Sigmoidna, tj. logistička aktivacijska funkcija uzima bilo koju stvarnu vrijednost kao ulaz i daje vrijednosti u rasponu od 0 do 1. Što je veći ulaz (pozitivniji), to će izlazna vrijednost biti bliža 1, dok što je manji unos (negativniji), to će izlaz biti bliži 0.

Funkcija sigmoidne, tj. logističke aktivacije jedna je od najčešće korištenih funkcija. Obično se koristi za modele u kojima moramo predvidjeti vjerojatnost kao izlaz. Budući da vjerojatnost bilo čega postoji samo između raspona od 0 do 1, sigmoid je pravi izbor zbog svog raspona [7].

Izgled funkcije možemo vidjeti na Slici 5.



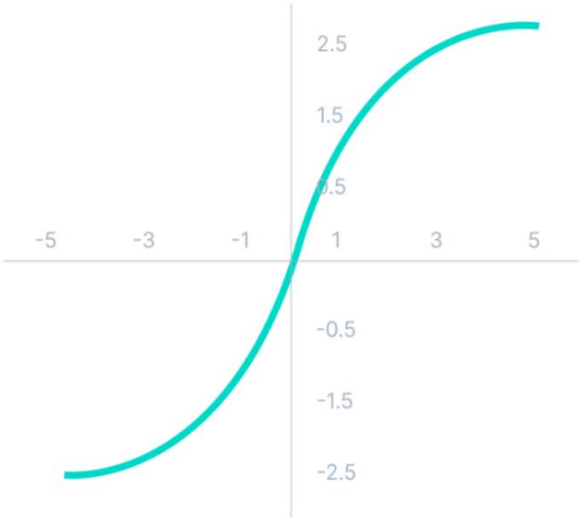
Slika 5. Sigmoidna funkcija [7]

Tanh funkcija, tj. hiperbolički tangens vrlo je slična sigmoidnoj aktivacijskoj funkciji i čak ima isti S-oblik s razlikom u izlaznom rasponu od -1 do 1. U tanhu, što je ulaz veći (pozitivniji), to je izlazna vrijednost bliža 1, dok što je ulaz manji (negativniji), to će izlaz biti bliži -1.

Prednosti korištenja ove funkcije su da je izlaz funkcije aktivacije tanh centriran na nulu; stoga možemo lako mapirati izlazne vrijednosti kao izrazito negativne, neutralne ili izrazito pozitivne.

Obično se koristi u skrivenim slojevima neuronske mreže jer su njegove vrijednosti između -1 do 1, srednja vrijednost za skriveni sloj je 0 ili vrlo blizu toga. Pomaže u centriranju podataka i olakšava učenje za sljedeći sloj [7].

Na Slici 6. vidimo grafički prikaz tanh funkcije.

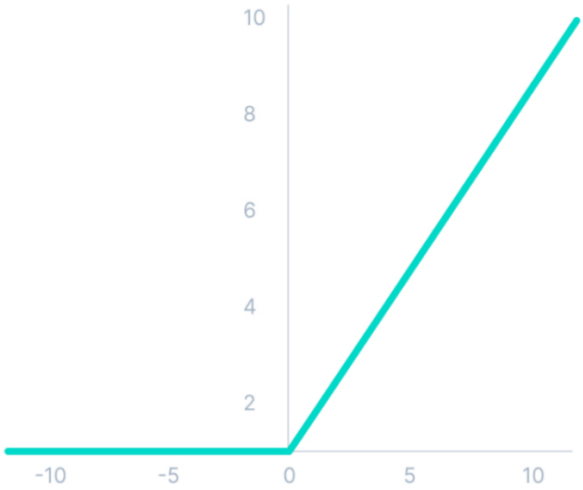


Slika 6. Tanh funkcija [7]

Iako ispravno linearno aktiviranje (engl. *Rectified Linear Unit*, krat. ReLU) daje dojam linearne funkcije. ReLU ima derivaciju u svim točkama osim točke 0 i dopušta širenje unatrag dok ga istovremeno čini računalno učinkovitim [7].

Glavna značajka ove funkcije je da ReLU funkcija ne aktivira sve neurone u isto vrijeme. Neuroni će biti deaktivirani samo ako je izlaz linearne transformacije manji od 0.

Grafički prikaz ReLU aktivacijske funkcije možemo vidjeti na Slici 7.



Slika 7. ReLU aktivacijska funkcija [7]

Budući da se aktivira samo određeni broj neurona, ReLU funkcija je računalno daleko učinkovitija u usporedbi sa sigmoidnom i tanh funkcijom [7].

2.8. Popularne metode učenja neuronskih mreža

Neuronske mreže mogu se klasificirati u različite vrste. U nastavku ćemo navesti najčešće vrste neuronskih mreža, način na koji rade i njihovu primjenu.

Feedforward neuronske mreže ili višeslojni perceptroni (engl. *multi-layer perceptrons*, krat. MLP) imaju ulazni sloj, jedan ili više skrivenih slojeva i izlazni sloa te koriste nelinearne aktivacijske funkcije. *Feedforward* model je najjednostavniji oblik neuronske mreže jer se informacije obrađuju samo u jednom smjeru. Iako podaci mogu proći kroz više skrivenih čvorova, uvijek se kreću u jednom smjeru, a nikada unatrag. Ovaj model je temelj za prepoznavanje govora i računalni vid te druge neuronske mreže.

Konvolucijske neuronske mreže (engl. *Convolutional neural networks*, krat. CNN) slične su *feedforward* neuronskim mrežama i obično se koriste za prepoznavanje slika, prepoznavanje uzoraka i računalni vid. Ove mreže koriste metode linearne algebre, posebno množenje matrica, za prepoznavanje uzoraka unutar slike. Razlikuju se od ostalih neuronskih mreža

svojim superiornim performansama s ulaznim signalima slike, govora ili zvuka. Imaju tri glavne vrste slojeva, a to su: konvolucijski sloj, sloj udruživanja i potpuno povezani sloj. U konvolucijskoj mreži prvi sloj je upravo konvolucijski sloj. Dodatni konvolucijski slojevi ili slojevi udruživanja mogu doći nakon prvog konvolucijskog sloja, a potpuno povezani sloj je zadnji. CNN postaje sve složeniji sa svakim slojem, identificirajući veće dijelove slike. Raniji slojevi fokusirani su na jednostavne značajke, kao što su boje i rubovi. Veće značajke ili oblici objekta počinju se prepoznavati kako slikovni podaci napreduju kroz CNN slojeve i na kraju se identificira željeni objekt [8].

Suprotno od *feedforward* neuronske mreže je ponavljajuća neuronska mreža (engl. *Recurrent neural networks,* krat. RNN), u kojoj se određeni putevi mijenjaju. Ponavljajuće neuronske mreže prepoznaju se po svojim povratnim vezama. Ti se algoritmi učenja primarno koriste za predviđanje budućih ishoda, kao što su predviđanja tržišta dionica ili predviđanja prodaje [9].

Ponavljajuća neuronska mreža je vrsta neuronske mreže gdje se izlaz iz prethodnog koraka unosi kao ulaz u trenutni korak. Kada je potrebno predvidjeti sljedeću riječ rečenice, prethodne se riječi moraju zapamtiti jer su u tradicionalnim neuronskim mrežama svi ulazi i izlazi neovisni jedni o drugima. Tako je nastao RNN čija je glavna i najvažnija značajka njegovo skriveno stanje, koje pamti neke informacije o nizu. Skriveno stanje se također naziva i stanje memorije budući da pamti prethodni unos u mrežu. Koristi iste parametre za svaki ulaz jer obavlja isti zadatak na svim ulazima ili skrivenim slojevima za dobivanje vrijednosti izlaza. Time se smanjuje složenost parametara, za razliku od drugih neuronskih mreža [10].

Metoda duge kratkoročne memorije (engl. *Long Short-Term Memory,* krat. LSTM) je vrsta RNN mreže koja sadrži „memorijsku ćeliju“ u kojoj čuva informacije dulje vrijeme. Sastoji se od skupa logičkih vrata: ulaznih vrata, izlaznih vrata i vrata za zaboravljanje (engl. *Input gate, output gate, forget gate*). Ulazna vrata odlučuju koji podatci će se čuvati u memoriji, izlazna vrata odlučuju koliko podataka se prenosi sljedećem sloju, a vrata za zaboravljanje kontroliraju brzinu odbacivanja pohranjenih podataka.

2.9. Ograničenja neuronskih mreža i dubokog učenja

Vijesti i istraživanja sve više pokazuju ograničenja dubokog učenja i neke od slabosti pristupa neuronskih mreža. Umjetne neuronske mreže zahtijevaju veliku količinu podataka za učenje, pa se mogu koristiti samo u određenim situacijama. Neuronske mreže su dobre u klasificiranju i grupiranju podataka, ali nisu dobre u dedukciji, zaključivanju i drugim scenarijima donošenja odluka ili učenja.

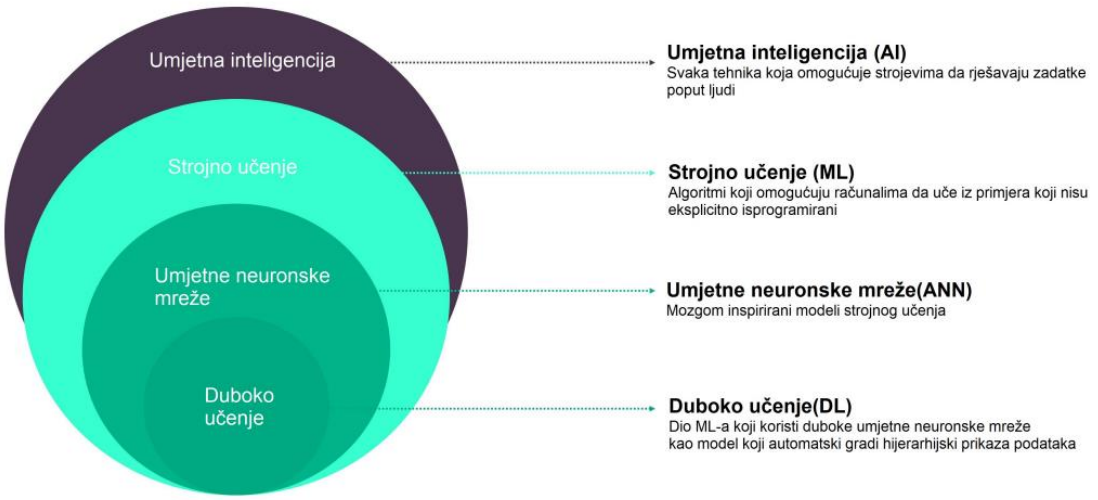
Slično kao što je teško razumjeti kako ljudski mozak donosi određene odluke, nemoguće je proučiti kako specifični ulazi u neuronsku mrežu dovode do rezultata na razumljiv ili transparentan način. Za aplikacije koje zahtijevaju uzročnu analizu ili niz uzročnih objašnjenja, neuronske mreže nisu održivo rješenje. U situacijama kada tumačenje treba poduprijeti važne odluke, metode "crne kutije" nisu uvijek prikladne ili dopuštene [11].

2.10. Usporedba strojnog i dubokog učenja

Budući da se duboko učenje i strojno učenje koriste naizmjenično, važno je uočiti nijanse između ta dva pojma. Strojno učenje, duboko učenje i neuronske mreže su podskup umjetne inteligencije. Neuronske mreže su zapravo podskup strojnog učenja, a duboko učenje je podskup neuronskih mreža.

Glavna razlika između strojnog i dubokog učenja je način na koji svaki algoritam uči. Duboko učenje većinom koristi označene skupove podataka, tj. nadzirano učenje za učenje svog algoritma, ali podatci ne moraju nužno biti označeni. Duboko učenje može uzeti podatke u njihovom izvornom obliku (npr. tekst ili slike) i automatski odrediti skup značajki po kojima se kategorije podataka razlikuju jedne od drugih. Time se eliminira dio ljudske intervencije i omogućuje korištenje većih skupova podataka [1].

Neuronske mreže sastoje se od ulaznog sloja, jednog ili više skrivenih slojeva i izlaznog sloja. "Duboko" u dubokom učenju odnosi se na broj slojeva u neuronskoj mreži. Neuronska mreža koja se sastoji od više od tri sloja, u koja spadaju ulazni i izlazni sloj, smatra se algoritmom dubokog učenja ili dubokom neuronskom mrežom. Neuronska mreža koja ima samo tri sloja je samo osnovna neuronska mreža [1].



Slika 8. Slojeviti prikaz umjetne inteligencije

3. Razvijanje neuronskih mreža uz pomoć MATLAB-a

Za izradu neuronske mreže u ovom radu ćemo koristit MATLAB razvojno okruženje točnije verziju MATLAB R2023a.

3.1. MATLAB

MATLAB (engl. *Matrix Laboratory*) je programsko okruženje za tehničko i znanstveno računanje koje omogućava izvođenje kompleksnih proračuna, vizualizaciju rezultata, izvođenje simulacija i programiranje. Također ima vlastiti programski jezik koji je jednostavan za korištenje i upotrebljava standardiziranu matematičku sintaksu te podržava objektno-orijentirani pristup. Dizajnirao ga je matematičar Cleve Moler krajem 70-ih godina na Sveučilištu u Meksiku za potrebe studenta, a kasnije se razvija u alat za znanstvenike i inženjere.

Korisničko sučelje MATLAB-a sastoji se od interaktivne konzole (engl. *Command window*) u koju se utipkavaju naredbe i vide njihovi rezultati, radnog prostora (engl. *Workspace*) s popisom svih varijabli i njihovih osnovnih svojstava, povijesti naredbi (engl. *Command history*) koja sadrži popis ranije izvršenih naredbi te trenutnog direktorija (engl. *Current folder*), tj. direktorija u kojem se nalaze programi dostupni za izvršavanje [12].

3.2. *GoogLeNet*

GoogLeNet je unaprijed obučena konvolucijska neuronska mrežu koja se sastoji od 144 sloja. S obzirom da ima više od 3 sloja radi se u mreži dubokog učenja te je potrebno koristiti *Deep Learning Toolbox*. Razvio ju je Google kako bi unaprijedio algoritam klasifikacije slika. Obučena neuronska mreža može klasificirati slike u 1000 kategorija kao što su tipkovnica, miš, olovka, itd.

GoogLeNet je dizajniran na način da se može ponovo uvježbati i primijeniti na nove podatke. Ovaj princip se zove prijenos učenja i koristit ćemo ga kako bi klasificirali slike u našem primjeru.

3.3. *Deep Learning Toolbox*

Deep Learning Toolbox nudi okvir za stvaranje i stavljanje u upotrebu dubokih neuronskih mreža koji uključuje algoritme, unaprijed obučene modele i aplikacije. Zadaci klasifikacije i

regresije mogu se provesti na slikovnim, vremenskim serijama i tekstualnim podacima korištenjem konvolucijskih neuronskih mreža i mreža duge kratkoročne memorije. Klasifikacija je proces pronalaženja funkcije koja pomaže u dijeljenju skupa podataka u klase na temelju različitih parametara. Zadatci klasifikacije se koriste za predviđanje vrijednosti koje se svrstavaju u klase kao što su: istina ili laž, spam ili nije spam, itd. Regresija je proces pronalaženja korelacija između zavisnih i nezavisnih varijabli. Koristi se kada je potrebno odrediti vrijednosti poput cijene, prihoda, itd.

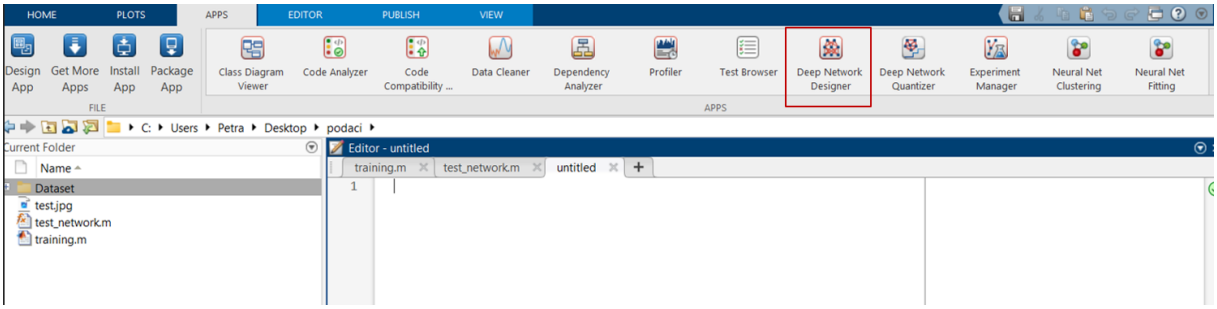
Deep Learning Toolbox nudi alate za svaku fazu tijeka rada dubokog učenja. Potrebno je procesirati podatke za sveobuhvatnu mrežnu obuku korištenjem funkcija i interaktivnih aplikacija. Nakon toga potrebno je uvesti obučene mreže iz MATLAB-a ili neke druge platforme. Slijedeći korak je stvaranje mreže pomoću naredbenih funkcija ili interaktivno korištenjem *Deep Network Designera*. Prije same obuke neuronske mreže treba odabrati najbolje opcije za učenje i formiranje mreže koristeći ugrađene značajke učenja ili prilagođavanjem petlje učenja. Također je moguće vizualizirati i ispitati ponašanje neuronske mreže tijekom i nakon učenja [13].

3.4. Razvijanje neuronske mreže koristeći *Deep Network Designer*

U ovom radu napravit ćemo neuronsku mrežu za klasifikaciju slika. Na temelju ulaznih podataka naučit ćemo mrežu da procijeni izlaznu vrijednost „X“ ili „Y“ za svaku sljedeću unesenu sliku.

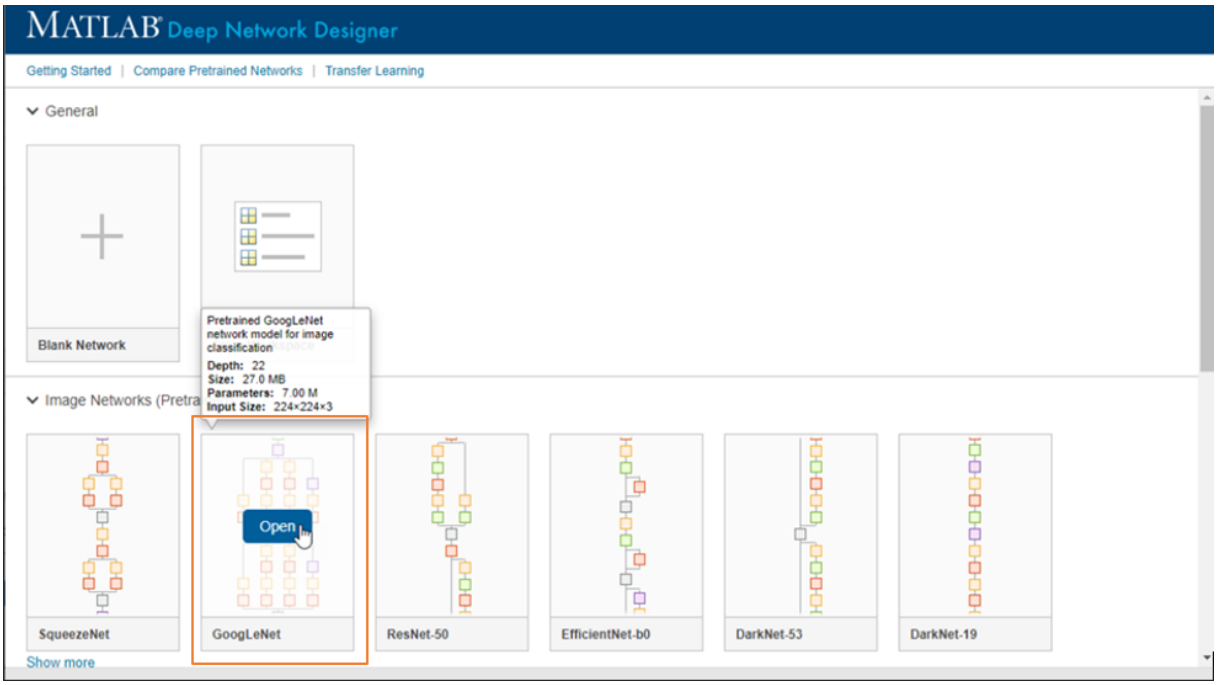
Kroz ovaj primjer ćemo pokazati kako koristiti *Deep Network Designer* za prilagodbu već obučene *GoogLeNet* mreže za klasificiranje nove zbirke slika. Ovaj se proces naziva prijenos učenja i obično je mnogo brži i lakši od obuke nove mreže, jer možemo primijeniti naučene značajke na novi zadatak koristeći manji broj slika za obuku [13].

Kako bi pokrenuli *Deep Network Designer* potrebno je na početnom MATLAB sučelju kliknuti na *„APPS“* u gornjem lijevom kutu na izbornoj traci te zatim izabrati *„Deep Network Designer“* kao što je prikazano na slici ispod.



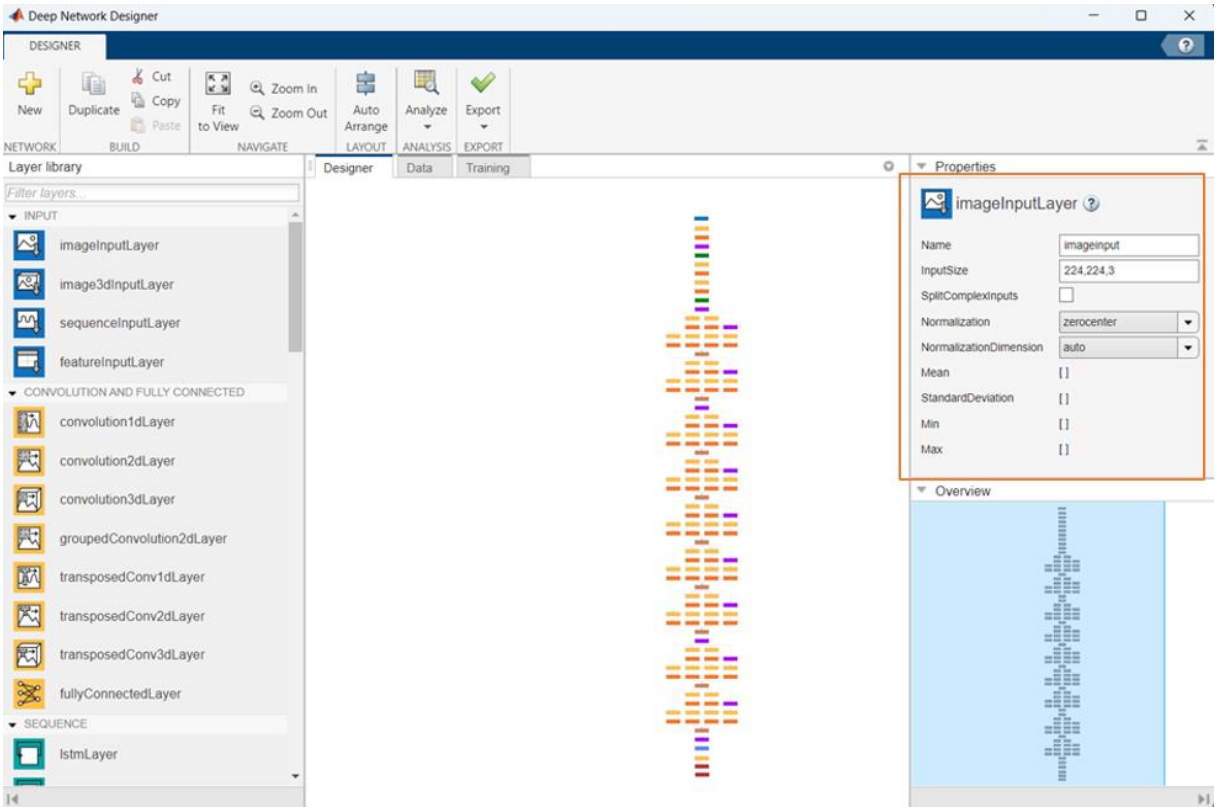
Slika 9. Početno MATLAB sučelje

Nakon što se otvori početni prozor *Deep Network Designera* potrebno je učitati unaprijed obučenu *GoogLeNet* mrežu.



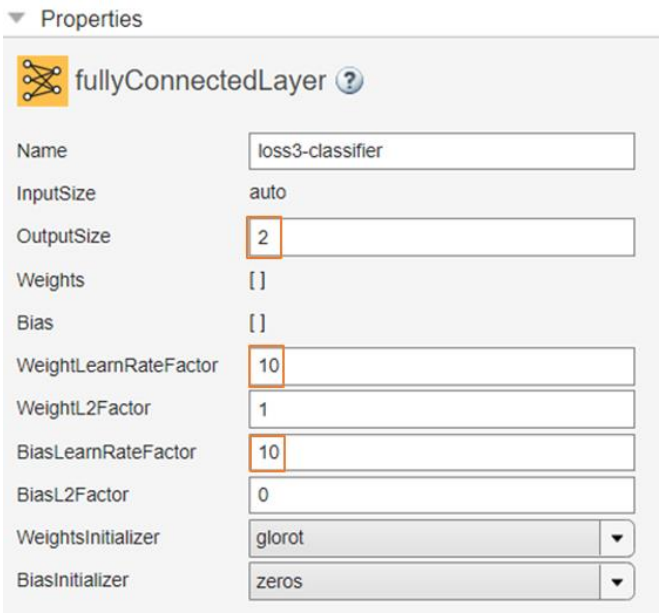
Slika 10. Početni prozor *Deep Network Designera*

Nakon što se učita *GoogLeNet* mreža prikaže se prozor sa Slike 11. Na lijevom dijelu vidimo popis svih slojeva mreže koje možemo po potrebi mijenjati. Na sredini se može vidjeti umanjeni prikaz cijele neuronske mreže. Ukoliko kliknemo na neki od slojeva sa slike desno će nam se prikazati svojstva odabranog sloja mreže. U našem primjeru možemo vidjeti svojstva za *„imageInputLayer“*. Iz Slike 11. možemo iščitati da je ulazna vrijednost za veličinu slike postavljena 224 x 224. Ta veličina daje najbolju ravnotežu između složenosti i točnosti prilikom klasifikacije. Veći format slika rezultirao bi većom točnosti, ali bi tada vrijeme učenja mreže bilo duže.

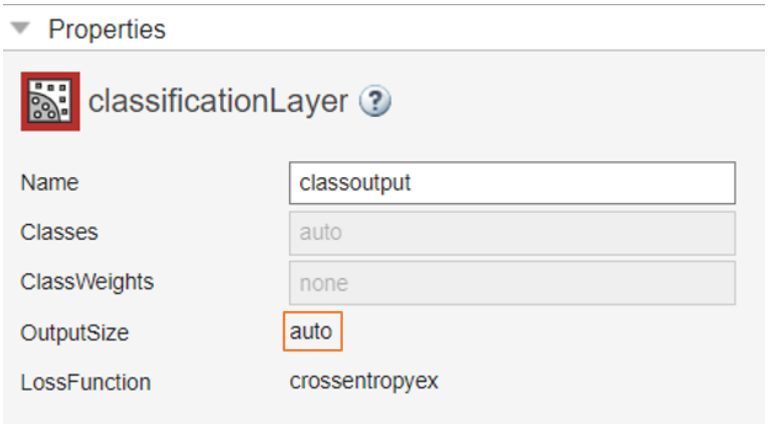


Slika 11. Prikaz slojeva u *Deep Network Designeru*

S obzirom da u našem slučaju imamo samo dvije izlazne vrijednosti „X“ i „Y“ potrebno je urediti određene slojeve u učitanoj *GoogLeNet* mreži. To napravimo tako što „dovučemo“ potreban sloj s lijeve strane na prikaz neuronske mreže koji se nalazi u sredini Slike 11. U našem slučaju potrebno je promijeniti vrijednosti sloja *„fullyConnectedLayer“* (prikazano na Slici 12.) i promijeniti sloj *„classificationLayer“*. Potrebno je promijeniti *„OutputSize“* sa defaultne vrijednosti tisuću na vrijednost dva jer u našem slučaju imamo dvije izlazne vrijednosti, *„WeightLearnRateFactor“* i *„BiasLearnRateFactor“* mijenjamo s defaultne vrijednosti jedan na vrijednost deset kako bi se povećala stopa učenja mreže. Kako smo promijenili broj izlaznih vrijednosti na dva u *„fullyConnectedLayer“* sloju taj podatak se automatski povuče kroz mrežu kada dovučemo novi *„classificationLayer“* kao što vidi na Slici 13.

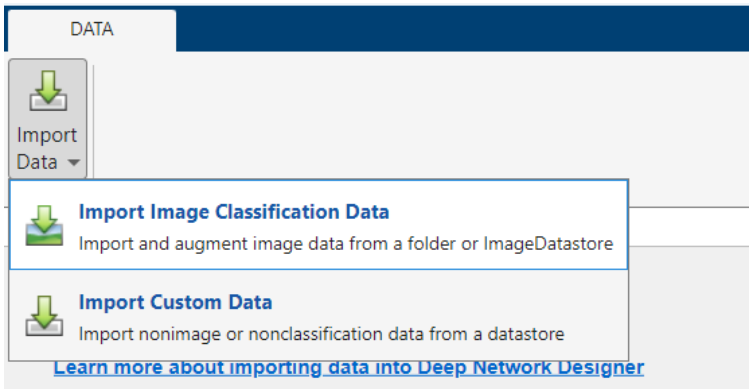


Slika 12. Nova svojstva „*fullyConnectedLayer*“ sloja



Slika 13. Novi *„classificationLayer“* sloj

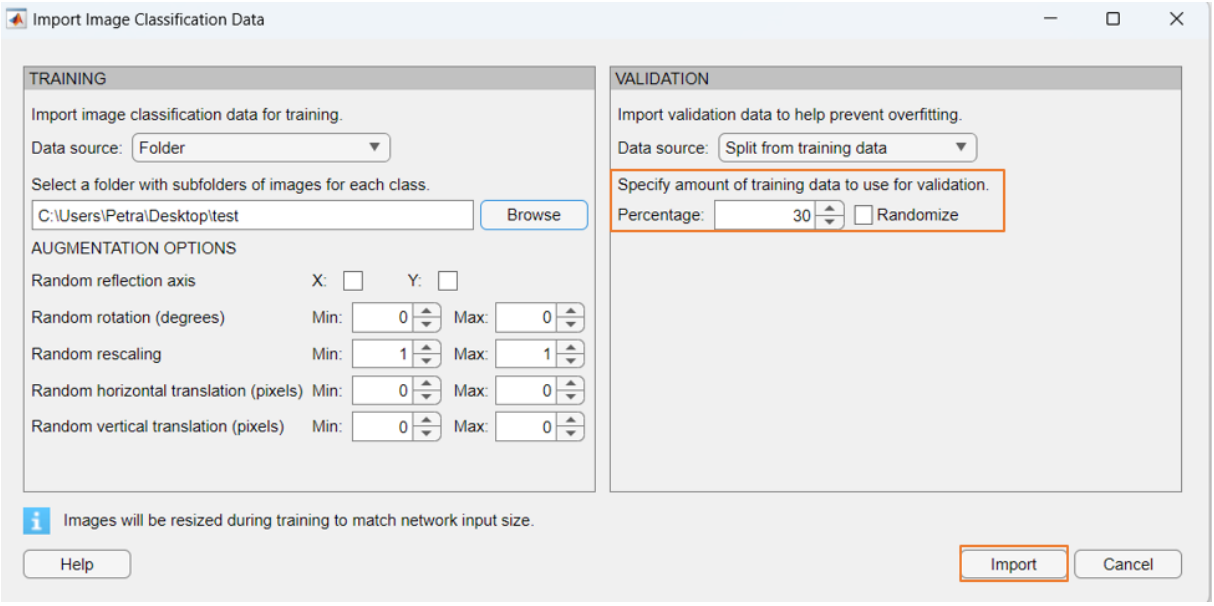
U sljedećem koraku dodajemo ulazne podatke klikom na „*Import Data“* te odabirom „*Import Image Classificaion Data“* kako bi nam ulazni podatci bili slike.



Slika 14. Opcije za odabir ulaznih podataka

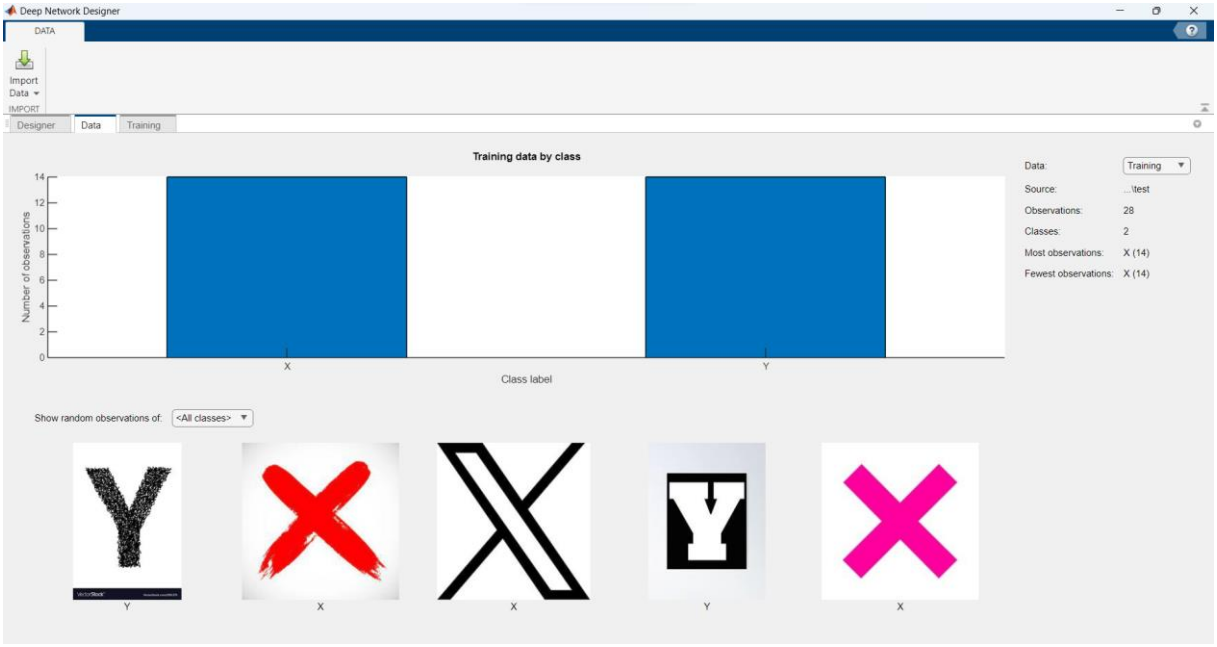
Sada se otvara sučelje za odabir datoteke s ulaznim podatcima. Klikom na tipku *„Browse“* potrebno je odabrati datoteku u kojoj nam se nalaze ulazni podaci. U našem slučaju datoteka

se nalazi na radnoj površini pod imenom „test“. Također je moguće promijeniti količinu podataka koja se koristi za učenje, a koja za validaciju neuronske mreže. Podatci za učenje su podatci na kojima se neuronska mreža uči, a podaci za validaciju služe za provjeru točnosti obučenog modela. U našem primjeru ćemo ostaviti defaultnu vrijednost gdje se 70% podataka koristi za učenje jer takva podjela daje najbolje rezultate.



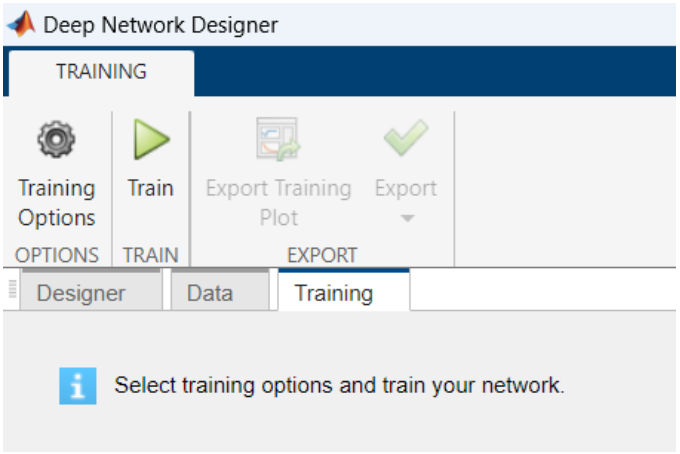
Slika 15. Sučelje za unos podataka

Nakon što uvezemo ulazne podatke preko prethodnog slučaja otvara se prozor sa Slike 16. Prije predaje ulaznih podataka potrebno je označiti koji podatci pripadaju klasi „X“, a koji klasi „Y“. To napravimo tako što u našoj datoteci „test“ napravimo dvije mape „X“ i „Y“ i stavimo odgovarajuće podatke u mapu. Sve slike moraju biti u istom formatu (u našem slučaju .jpg) jer se u suprotnom mogu pojaviti greške. Sa slike se jasno vidi da su ulazni podatci unaprijed razvrstani u dvije klase, klasu „X“ i klasu „Y“ te možemo vidjeti slučajne primjere iz obje klase.



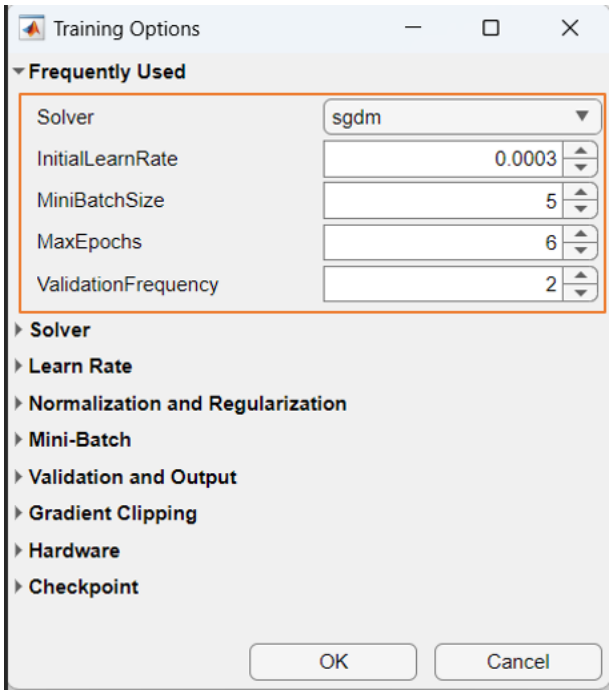
Slika 16. Prikaz ulaznih podataka

Sljedeći korak je učenje neuronske mreže. Prije nego što počnemo s učenjem mreže potrebno je prilagoditi opcije učenja klikom na *„Training Options“*.



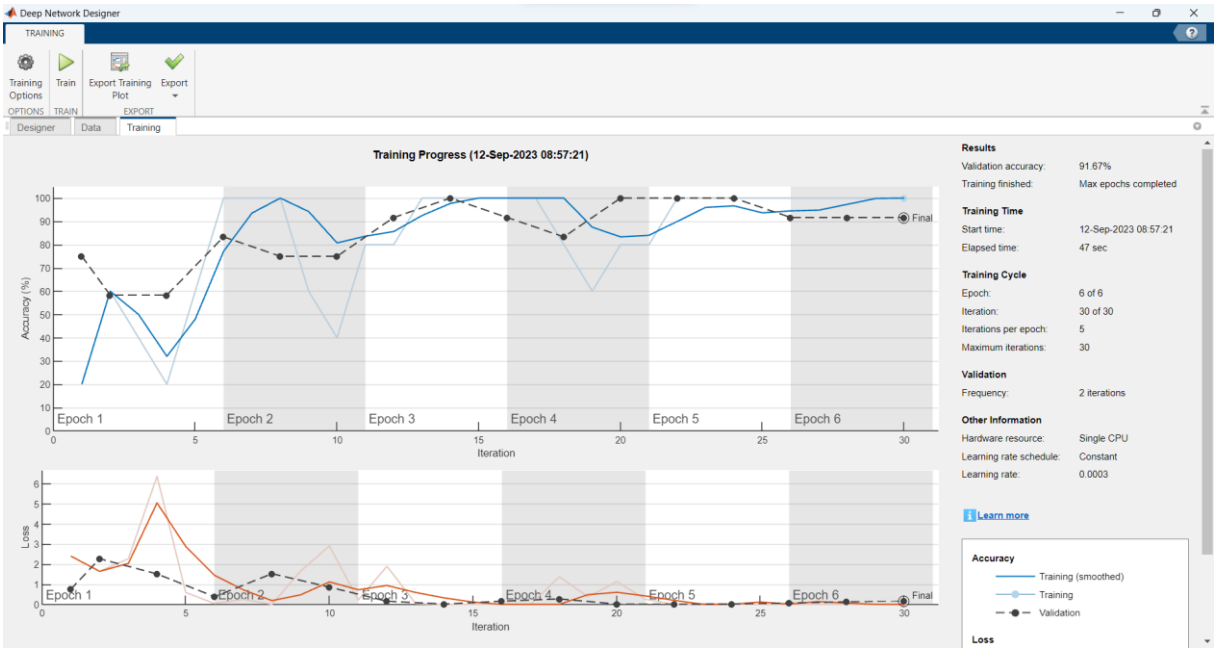
Slika 17. Sučelje za učenje neuronskih mreža

U našem slučaju potrebno je promijeniti vrijednosti *„InitialLearnRate“* koja kontrolira brzinu kojom model uči na vrijednost 0.0003 jer ta vrijednost daje najbolje rezultate učenja. Vrijednost *„MiniBatchSize“* je fiksni broj primjera obuke koji je manji od stvarnog skupa podataka. S obzirom da u našem primjeru imamo 40 ulaznih vrijednosti *„MiniBatchSize“* ćemo postaviti na pet. *„Epoch“* predstavlja potpuni prolaz skupa podataka za obuku kroz algoritam. U većini primjera jedanaest je idealna veličina za obuku mreže, ali kako u našem primjeru imamo mali broj ulaznih podataka veličinu *„MaxEpochs“* postavljamo na šest. *„ValidationFrequency“* postavljamo na vrijednost dva jer nam je vrijednost *„MiniBatchSize“* mala pa je potrebno smanjiti i *„ValidationFrequency“*.



Slika 18. Nove vrijednosti za učenje podataka

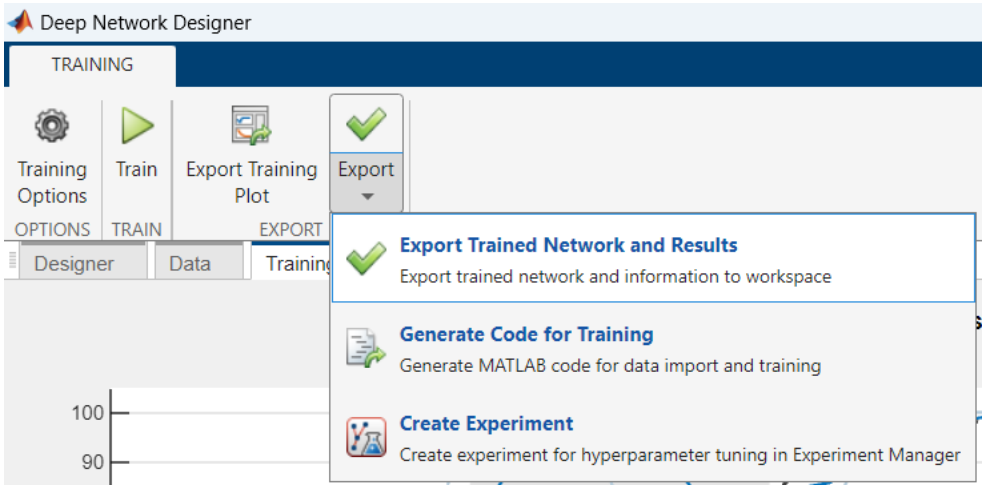
Nakon što smo promijenili opcije učenja naše neuronske mreže možemo početi s njenim učenjem klikom na tipku *„Train“* (Slika 17). Kao rezultat učenja dobili smo da je *„Validation accuracy“*, tj. točnost modela 91.67% što je dobar rezultat.



Slika 19. Rezultati učenja neuronske mreže

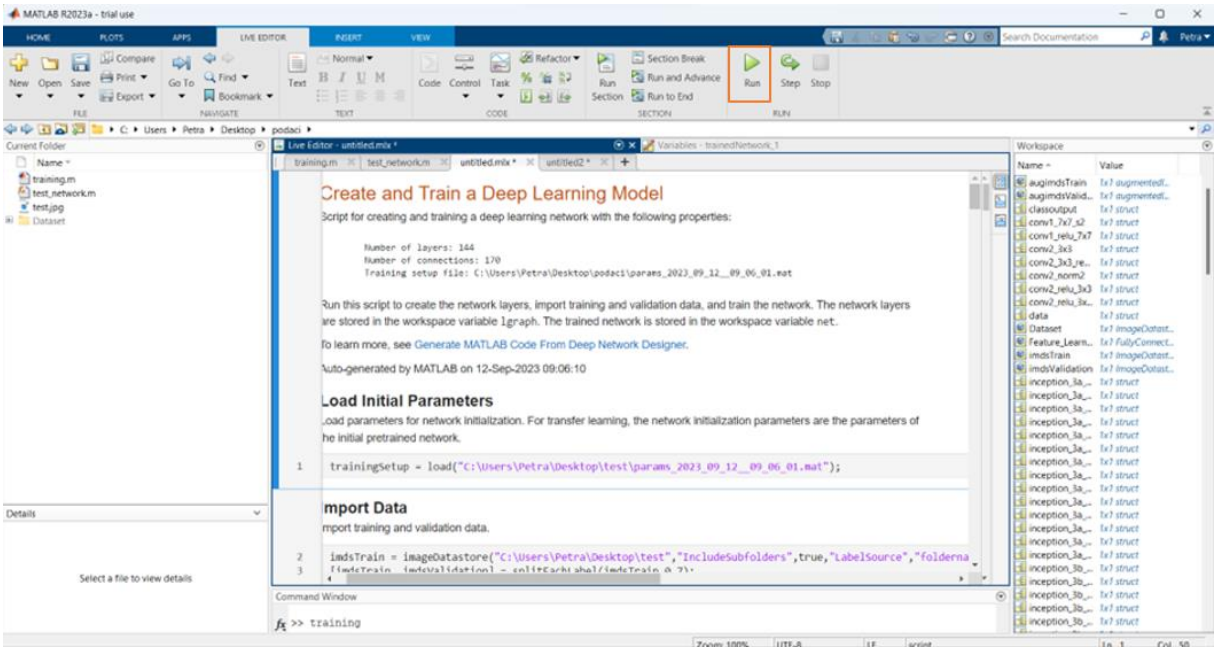
Ukoliko sada zatvorimo *„Deep Network Designer“* naš model bi bio izgubljen te bi trebali ponoviti cijeli prethodni proces. Iz tog razloga potrebno je izvući MATLAB kod kako bi sačuvali obučenu neuronsku mrežu. Kako bi dobili MATLAB kod za učenje potrebno je

kliknuti na tipku *„Export“*, zatim pritisnuti na *„Export Trained Network and Results“* te nakon toga pritisnuti *„Generate Code for Training“*.



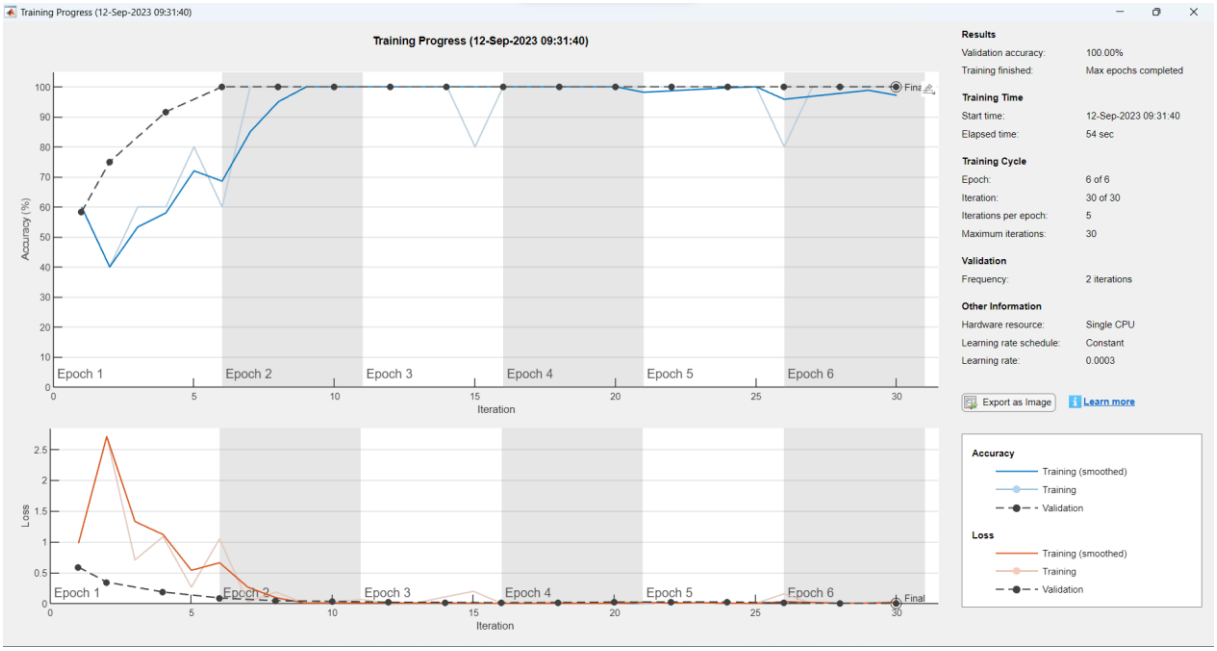
Slika 20. Opcije za dobivanje MATLAB koda

Nakon što smo napravili prethodni korak možemo vidjeti kod u MATLAB okruženju te ga pokrenuti.



Slika 21. Generirani kod u MATLAB *Live Editoru*

Kod s prethodne slike moguće je pokrenuti pritiskom na tipku *„Run“* te možemo i preko generiranog koda ponovno učiti neuronsku mrežu na istom primjeru. S obzirom da istu mrežu ponovno učimo sada dobijemo veću vrijednost točnosti učenja od 100% kao što možemo vidjeti na Slici 22.

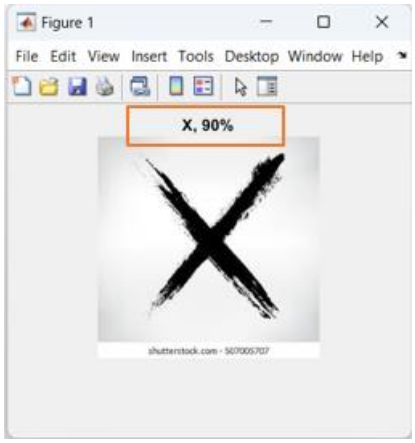


Slika 22. Učenje neuronske mreže pomoću prethodno generiranog MATLAB koda

Za testiranje naše neuronske mreže na primjeru potrebno je napisati kod sa sljedeće slike pomoću kojeg učitavamo novu ulaznu vrijednost, tj. sliku i postavljamo je da bude u skladu s ulaznim vrijednostima naše neuronske mreže iščitanim sa Slike 11. Potrebno je unijeti novu sliku koja se nije koristila za učenje neuronske mreže.

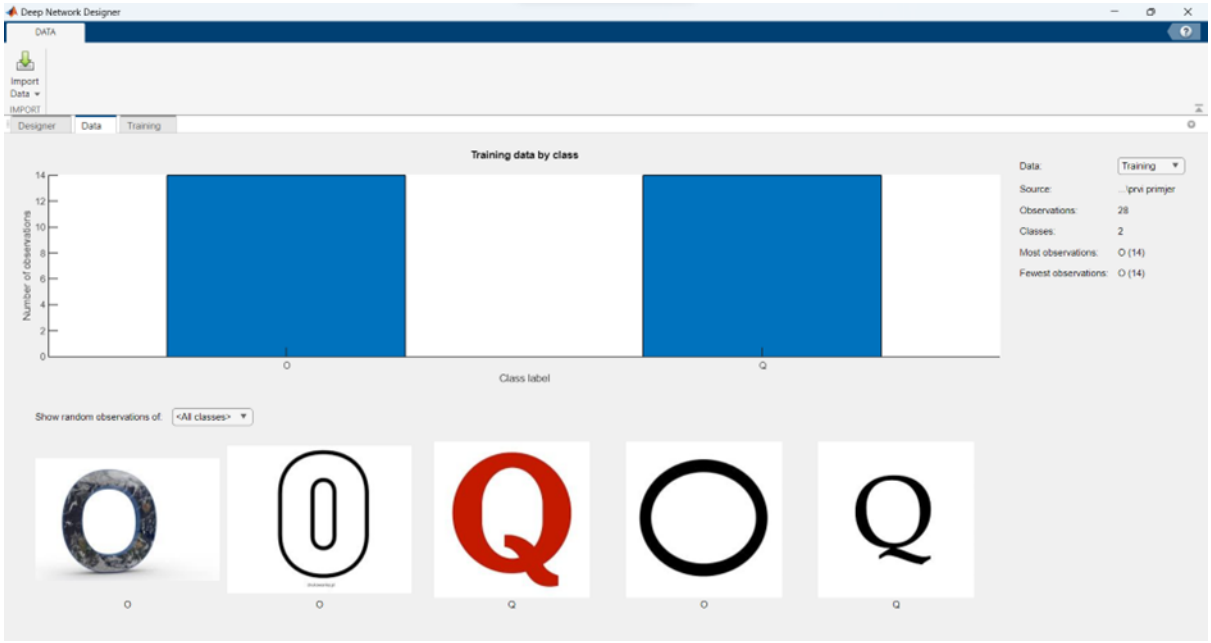
Slika 23. Kod za testiranje

Rezultati testiranja neuronske mreže iz ovog primjera mogu se vidjeti na Slici 24. Ulaznu sliku pod nazivom „testna.jpg“ prikazanu ispod obučena mreža je s 90% vjerojatnosti označila kao dio klase „X“.



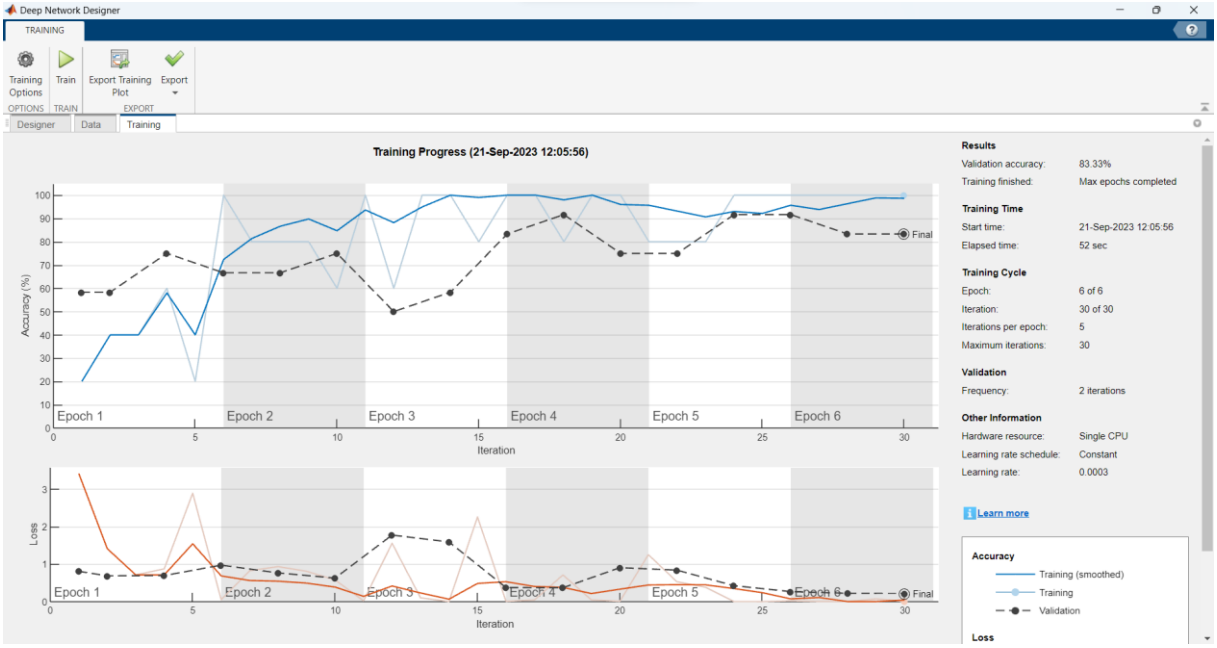
Slika 24. Rezultat testiranja

Napravit ćemo još jednu identičnu neuronsku mrežu, ali ćemo koristiti druge ulazne podatke. U ovom primjeru uzet ćemo slova „O“ i „Q“ te ispitati kolika je točnost učenja kada su ulazne vrijednosti slične. Na Slici 25. možemo vidjeti unesene podatke.



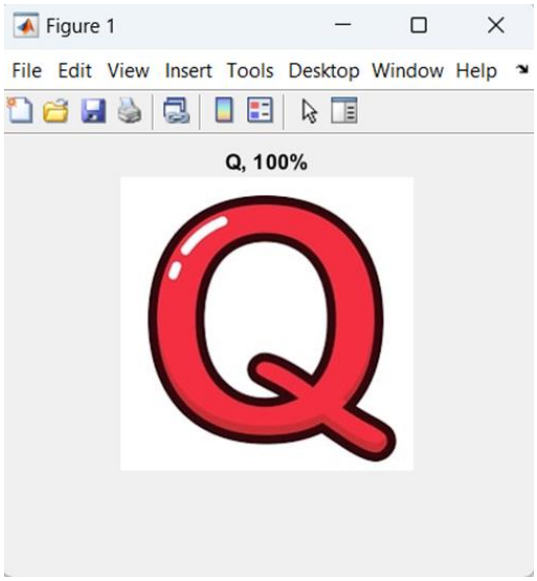
Slika 25. Prikaz ulaznih podataka za primjer „O“ i „Q“

Nakon što naučimo neuronsku mrežu na novom skupu podataka možemo primijetiti točnost od 83.33%. Ovi rezultati nam pokazuju da je *GoogLeNet* od početka precizno naučena čak i na malom skupu sličnih podataka (i u ovom primjeru imali smo 40 ulaznih vrijednosti), tj. dobro je naučila kako prepoznavati karakteristike u podacima i donositi točne odluke na temelju istih.



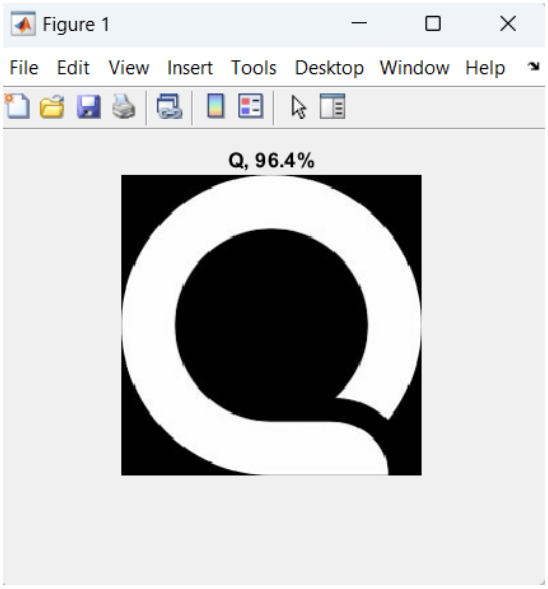
Slika 26. Rezultati učenja za primjer „O“ i „Q“

Također smo ponovili postupak za testiranje neuronske mreže na primjeru koji nije korišten pri učenju neuronske mreže. U ovom slučaju mreža je sa 100% sigurnosti odredila da slijedeća slika pripada klasi „Q“.



Slika 27. Rezultati testiranja primjera slike „Q“

Mrežu ćemo testirati i na primjerima u kojima slovo nije jasno napisano. U slučaju sa Slike 28. i sa Slike 29. slovo nema oštre crte. U slučaju sa Slike 28. mreža je sa 96.4% sigurnosti odredila da slika pripada klasi „Q“. U slučaju sa Slike 29. na kojem je prikazano animirano slovo Q mreža je također točno odredila da slika pripada klasi „Q“ u ovom slučaju sa 96% sigurnosti.



Slika 28. Rezultati testiranja na nejasno napisano slovo

Slika 29. Rezultati testiranja na animiranom slovu

Iz prethodnih primjera jasno vidimo da je mreža u svakom slučaju točno klasificirala slike. Na primjeru gdje je slovo bilo najjasnije napisano rezultati su bili najbolji. Za zadnja dva primjera sigurnost procjene se smanjila, ali je još uvijek ostala visoka na čemu vidimo da je mreža dobro naučena za izvršiti klasifikaciju.

4. ZAKLJUČAK

U ovom radu je prikazana primjena obučene *GoogLeNet* neuronske mreže na primjeru klasifikacije slika. U našem primjeru koristili smo prijenos učenja kako bi ulazne vrijednosti klasificirali kao dio klase „X“ ili dio klase „Y“. Mreža je obučena na malom skupu od 40 ulaznih podataka, ali su rezultati učenja bili dobri upravo zbog korištenja tehnike prijenosa učenja. Našu neuronsku mrežu izradili smo uz pomoć MATLAB okruženja korištenjem aplikacije *Deep Network Designer*.

Prije same obuke neuronske mreže objasnili smo što su neuronske mreže te što je sve potrebno za njenu izradu. Prije same obuke potrebno je prikupiti ulazne podatke, podijeliti ih u klase te dodati u neuronsku mrežu. Nadalje, potrebno je promijeniti unaprijed zadane vrijednosti na kojima je *GoogLeNet* neuronska mreža originalno obučena. Naknadno smo preko sučelja obučili neuronsku mrežu s novim vrijednostima i prikazali rezultate obuke. Svaki od navedenih koraka prikazan je u ovom radu.

Kroz ovaj rad smo pokazali da je MATLAB razvojno okruženje praktičan alat izrade i učenja neuronske mreže. *Deep Network Designer* jednostavan je za korištenje te nije potrebno prethodno znanje MATLAB-a kako bi se izradila neuronska mreža. Ukoliko imamo ulazni skup podataka nije potrebno napredno znanje iz MATLAB-a kako bi stvorili neuronsku mrežu kojom ćemo olakšati svakodnevni život.

LITERATURA

[1] »ibm,« ibm, [Mrežno]. Available: https://www.ibm.com/topics/machine-learning. [Pokušaj pristupa 21 Ožujak 2023].

[2] MathWorks, »mathworks,« mathworks, [Mrežno]. Available: https://www.mathworks.com/discovery/deep-learning.html. [Pokušaj pristupa 30 Srpanj 2023].

[3] A. Gupta, »Aiche,« Aiche The global home of chemical engineers, Lipanj 2018. [Mrežno]. Available: https://www.aiche.org/resources/publications/cep/2018/june/introduction-deep-learning-part-1?gclid=Cj0KCQjwn9CgBhDjARIsAD15h0Bw2of1CLqJmVnT6h8FK2Cd14CBDBvQknbNkR0r2jt1IVmyQrHnr1QaApIJEALw_wcB. [Pokušaj pristupa 30 Srpanj 2023].

[4] A. Lopac Groš i O. Meštrović, »Biologija 8,« CARNET, [Mrežno]. Available: https://edutorij.e-skole.hr/share/proxy/alfresco-noauth/edutorij/api/proxy-guest/3b8a4b4e-84b0-4580-aa6f-e38efe028ed9/biologija-8/m03/j01/index.html. [Pokušaj pristupa 25 Srpanj 2023].

[5] M. Banoula, »simplilearn.com,« simplilearn, 10 Svibanj 2023. [Mrežno]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron#biological_neuron. [Pokušaj pristupa 25 Srpanj 2023].

[6] »Elements of AI,« Elements of AI, [Mrežno]. Available: https://course.elementsofai.com/hr/5/2. [Pokušaj pristupa 15 Kolovoz 2023].

[7] P. Baheti, »v7labs,« Microsoft, 27 Svibanj 2021. [Mrežno]. Available: https://www.v7labs.com/blog/neural-networks-activation-functions. [Pokušaj pristupa 15 Kolovoz 2023].

[8] »ibm,« ibm, [Mrežno]. Available: https://www.ibm.com/topics/convolutional-neural-networks. [Pokušaj pristupa 18 Kolovoz 2023].

[9] »ibm,« ibm, [Mrežno]. Available: https://www.ibm.com/topics/neural-networks. [Pokušaj pristupa 17 Kolovoz 2023].

[10] »Geeks for geeks,« Geeks for geeks, [Mrežno]. Available: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/. [Pokušaj pristupa 18 Kolovoz 2023].

[11] K. Walch, »enterprise ai,« techtarget, 17 Rujan 2021. [Mrežno]. Available: https://www.techtarget.com/searchenterpriseai/feature/How-neural-network-training-methods-are-modeled-after-the-human-brain. [Pokušaj pristupa 17 Kolovoz 2023].

[12] G. Erceg, *MATLAB,* Split: PMFST, 2012.

[13] »MathWorks,« MathWorks, [Mrežno]. Available: https://www.mathworks.com/help/deeplearning/gs/get-started-with-deep-network-designer.html. [Pokušaj pristupa 7 Rujan 2023].

PRILOZI

Popis slika

IZJAVA O AUTORSTVU I IZVORNOSTI RADA

kojom ja Petra Majstorović, 0177057002, izjavljujem pod punom moralnom odgovornošću da:

- sam završni rad pod naslovom: Primjena neuronskih mreža na klasifikaciju slika pomoću MATLAB-a, na studiju: Primijenjeno/poslovno računarstvo, izradio samostalno, pod mentorstvom prof. dr. sc. Martina Lazara.
- Sam u izradi koristio navedenu literaturu i pri tome se pridržavao etičkih standarda u citiranju i korištenju izvora te niti jedan dio rada nije izravno preuzet iz tuđih radova.
- Sam suglasan da se sadržaj moga rada trajno pohrani i objavi u Repozitoriju Sveučilišta u Dubrovniku te se time, putem interneta učini javno i bez naknade dostupan svima**.**
- sadržaj moga rada u potpunosti odgovara sadržaju obranjenog i eventualno nakon obrade uređenog rada.
- sam prilikom korištenja slika s interneta poštovao autorska prava

Ime i prezime studenta:
Petra Majstorović

Potpis

U Dubrovniku, 25.9.2023.