

Veštačka Inteligencija II Projekat

MLProгноza

Prognoziranje vremena korišćenjem
neuronskih mreža

Profesor: Prof. dr. Leonid Stoimenov

Studenti (članovi tima Biser 2.2):

Nikola Lugić, 15193

Petar Maksimović, 15197

Jovan Pešić, 15301

Elektronski Fakultet Niš, Februar 2018.

Uvod i opis problema

Problem kojim se bavimo je prognoziranje vremena, tj. temperature, u zadatom vremenskom periodu za zadanu lokaciju u Srbiji korišćenjem mašinskog učenja - konkretno neuronskih mreža. Iz seta meteoroloških podataka se, na osnovu više parametara, određuje temperatura za taj period korišćenjem neuronske mreže sa povratnom spregom, uz korišćenje Rprop heuristike. Izlaz programa je poređenje izmerene i „prognozirane“ temperature, prikazano grafikom ovih vrednosti.

Model podataka

Podaci čiju obradu vršimo su podaci sa meteoroloških stanica u Srbiji koji se nalaze na FTP serveru na adresi <ftp://ftp.ncdc.noaa.gov/pub/data/noaa/> u sklopu *Integrated Surface Data* dataset-a. Ovi podaci sadrže podatke o izmerenoj temperaturi, vazдушnom pritisku, pravcu i brzini vetra, padavinama itd., koji se pamte u ISH formatu, pa ih je potrebno parsirati u čitljiv oblik. U tu svrhu smo koristili već dostupni ISH Parser pisan u Javi i preuzet sa iste lokacije sa koje uzimamo same podatke. Svaki ISH fajl beleži podatke za jednu vremensku stanicu tokom cele godine. Nakon preuzimanja željenih podataka sa FTP servera, oni se parsiraju u CSV fajlove, a zatim se iz njih čitaju potrebni podaci i smeštaju u prilagođeni model podataka koji se koristi kao ulaz neuronske mreže. Izgled tog modela prikazan je na slici ispod.

```
public DateTime Time { get; set; } // vreme merenja
public int WindSpeed { get; set; } // brzina vetra u m/s
public int Temperature { get; set; } // temperatura u C
public int DewPointTemperature { get; set; } // temperatura kondenzacije u C
public float AtmosphericPressure { get; set; } // atmosferski pritisak
public float[] Precipitation { get; set; } // padavine u mm poslednjih 1/6/12/24 sata
public float SnowDepth { get; set; } // dubina snega u mm
```

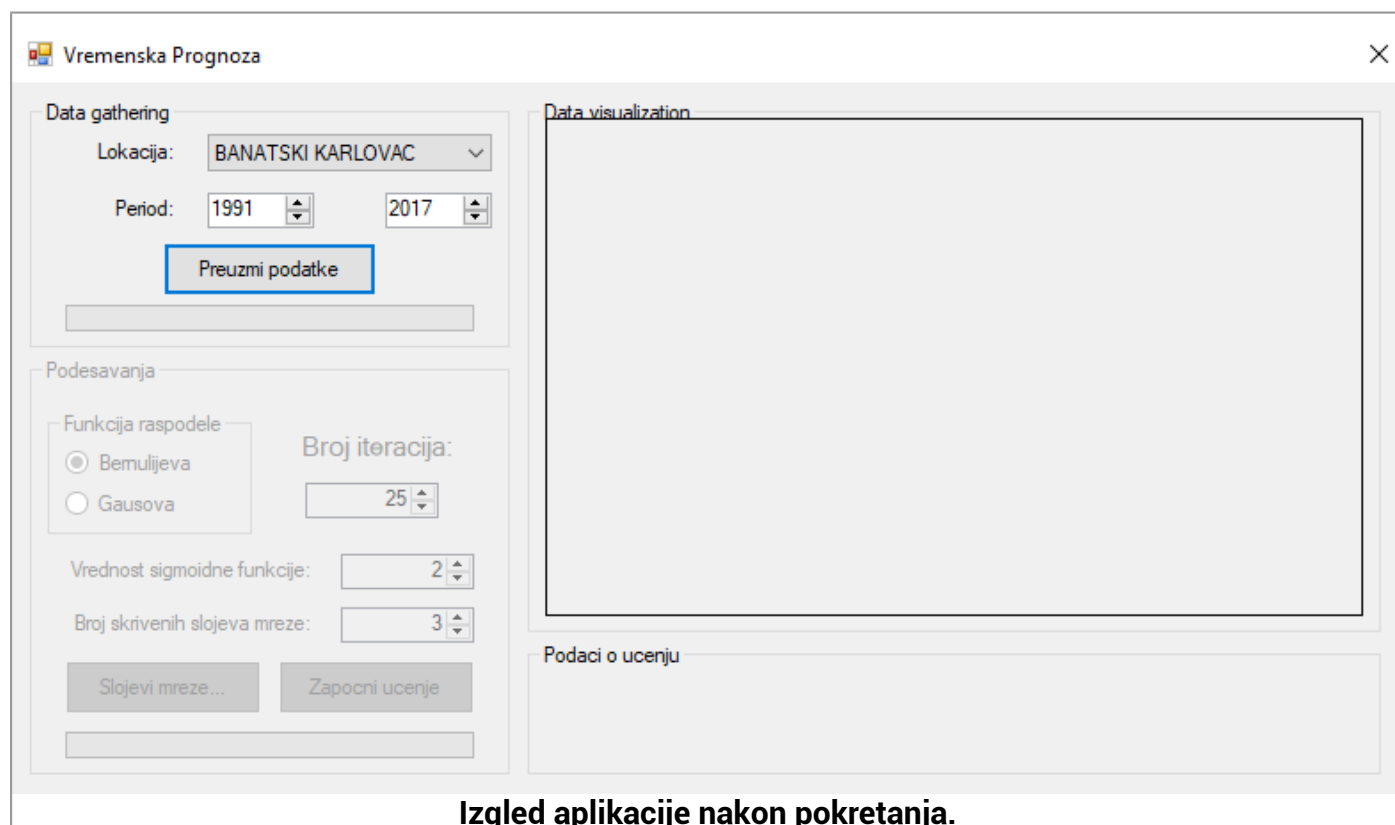
Izgled modela podataka koji koristi neuronska mreža.

Opis rešenja - aplikacije MLPrognoza

Aplikacija kojom demonstriramo rešenje ovog problema je *Windows Forms* aplikacija koja se sastoji iz dela interfejsa za prikupljanje podataka, dela za podešavanje parametara neuronske mreže i učenja, kao i dela za prikaz podataka na grafiku. Koristimo gotovu implementaciju neuronske mreže iz C# biblioteke za mašinsko učenje *Accord.NET*. Rešenje se, pored same aplikacije, sastoji iz još dve bitne komponente: komponente za preuzimanje i prevođenje podataka, kao i komponente za parsiranje podataka iz ISH formata u čitljivi CSV format. Prva od ove dve komponente je realizovana sa naše strane u projektu *MLPrognoza.Data*, dok je druga već pomenuti parser pisan u Javi, koji naša aplikacija poziva za konverziju svakog fajla.

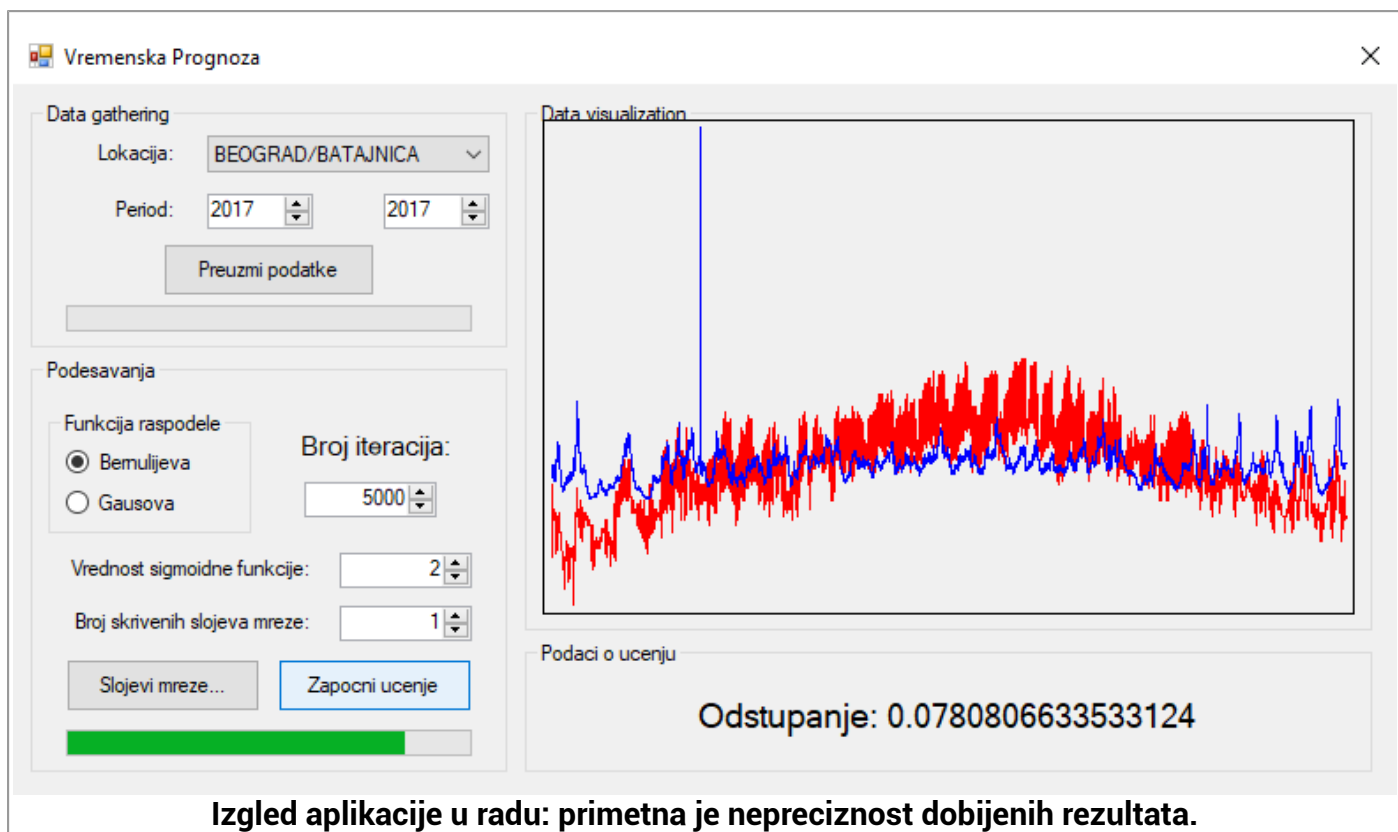
Prilikom startovanja aplikacije, korisniku je prikazana forma sa dostupnim kontrolama za odabir vremenske stanice u Srbiji i opsega godina koje ulaze u set podataka za obradu. Nakon pritiska na dugme „Preuzmi podatke“ prvo se poziva metoda *DownloadWeatherData* iz klase *FileDownloader*, koja na osnovu odabrane vremenske stanice i opsega godina preuzima odgovarajuće ISH fajlove. Zatim se preuzeti fajlovi obrađuju ISH parserom, koji se poziva kao Java aplikacija metodom *ConvertISHFile* klase *ISHParser*, čime se dobijaju CSV fajlovi koji sadrže željene podatke. U ovom koraku će se *Command Prompt* prozor pojaviti više puta, zbog poziva Java programa. Za čitanje i

prevođenje ovih podataka u model koristili smo biblioteku *FileHelpers*, koja na jednostavan način omogućava prevođenje vrsta nekog CSV fajla u niz objekata odgovarajuće klase. Ta konverzija se obavlja u klasi *WeatherData* koja ima 3 metode: *GetWeatherStationData*, koja vraća podatke o vremenskim stanicama (koristi se za učitavanje stanica koje se mogu odabrati), *GetWeatherRecordsData*, koja vraća „sirove“ podatke pročitane iz parsiranih fajlova, kao i *GetWeaterModelData*, koja od tih „sirovih“ podataka kreira set podataka koji se može koristiti kao ulaz neuronske mreže. Poslednje dve metode odavde čine dva poslednja koraka u konverziji podataka, i nakon njihovog izvršenja imamo podatke koji će biti ulaz neuronske mreže. Prikaz aplikacije nakon startovanja može se videti na slici ispod.



Izgled aplikacije nakon pokretanja.

Sledeći deo aplikacije je deo za podešavanje parametara neuronske mreže i učenja. Ovde se bira funkcija raspodele, vrednost sigmoidne funkcije, broj skrivenih slojeva mreže (uz mogućnost odabira broja čvorova u svakom sloju), kao i broj iteracija učenja mreže. Nakon odabira ovih podataka, klikom na dugme „Započni učenje“, inicijalizuje se neuronska mreža na osnovu odabranih podešavanja i započinje proces učenja koji se može pratiti prikazom podataka na grafiku, kao i podataka o trenutnoj iteraciji i grešci aproksimacije, koji se nalaze ispod grafika. Crvenom linijom iscrtava se grafik izmerenih vrednosti, dok je plavom iscrtan grafik vrednosti dobijenih sa izlaza neuronske mreže u poslednjoj iteraciji. Izgled aplikacije u radu prikazan je na slici na sledećoj strani.



Zaključak

Ova aplikacija pruža uvid u samo jednu od brojnih oblasti primene mašinskog učenja, sa fokusom na praktični razvoj uz korišćenje gotovih rešenja. Dobijeni rezultati nisu ni približno savršeni jer bi za to bilo potrebno previše podataka i procesorskog vremena. Ipak, ova aplikacija, koja je u isto vreme jednostavna za implementaciju i poučna može biti osnova za neko ozbiljnije izučavanje mašinskog učenja, ili pak organizaciju dela laboratorijskih vežbi/domaćih zadataka na predmetu Veštačka inteligencija u bližoj ili daljoj budućnosti.