

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6346

# **Konvolucijski modeli za jednooku predikciju dubine scene**

Filip Oreč

Zagreb, svibanj 2019.

*Umjesto ove stranice umetnite izvornik Vašeg rada.  
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*

*Hvala*

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Umjetna neuronska mreža</b>	<b>2</b>
2.1. Umjetni neuron . . . . .	2
2.1.1. Prijenosne funkcije . . . . .	3
2.2. Arhitektura umjetne neuronske mreže . . . . .	6
2.3. Učenje umjetne neuronske mreže . . . . .	7
2.3.1. Funkcija gubitka . . . . .	7
<b>3. Konvolucijska neuronska mreža</b>	<b>8</b>
<b>4. Zaključak</b>	<b>9</b>
<b>Literatura</b>	<b>10</b>

# 1. Uvod

Predikcija dubine scene je još davno poznati problem. Konvencionalni prikazi kao što su slika ili video prikazuju trodimenzionalni svijet u dvije dimenzije. Naime, time gubimo informaciju o trećoj dimenziji koja sadrži informaciju o dubini scene. Iako je dvodimenzionalna reprezentacija dovoljna u većini primjena, nekad nam je potrebna trodimenzionalna reprezentacija. Percepcija dubine proizlazi iz raznih znakova ili naznaka o dubini. Obično se dijele na binokularne znakove koji sadrže informacije u tri dimenzije i mogu se vidjeti s dva oka i monokularne znakove koji sadrže informacije u dvije dimenzije i mogu se vidjeti s jednim okom.

Tijekom godina razvile su se razne tehnike za predikciju dubine poput stereoskopije koja se oslanja na binokularne znakove. Korištenjem dvije slike iste scene koje su dobivene iz malo drugačijih kuteva, moguće je izračunati udaljenost od objekta. Aplikacijama koje uključuju razumijevanje scene, 3D modeliranje i slično jako je bitna informacija o dubini kad gore navedena organizacija podataka i tehnike nisu dostupni. U tom slučaju koristi se jednooka predikcija dubine scene koja predstavlja loše postavljen (engl. *ill-posed*) problem, jer iz jedne dvodimenzionalne slike može nastati beskonačno mnogo različitih trodimenzionalnih scena. Za ljude ovo ne predstavlja veliki problem, jer možemo jako dobro iskoristiti monokularne znakove, ali za računala predstavlja ogroman problem za riješiti s velikom preciznosti i malom uporabom resursa. Zbog navedenog razloga, u zadnje vrijeme upotrebljavaju se konvolucijski modeli koji uče odnos između piksela boje i dubine, što je i predmet ovog rada. Detaljnije ću obraditi konvolucijsku neuronsku mrežu predloženu u [2], te ju implementirati i primijeniti na problem jednooke predikcije dubine scene.

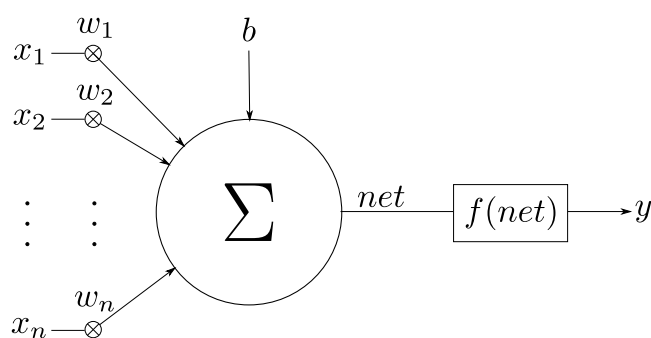
## 2. Umjetna neuronska mreža

Nastanak umjetnih neuronskih mreža u početku je bio inspiriran biološkim neuronima i neuronskim mrežama, ali daljnjim razvojem su se odvojile od biološke povezanosti i postale stvar inženjerstva. Definiraju se procesne jedinice zvane umjetni neuroni koji se međusobno povezuju te grade umjetne neuronske mreže.

Ovaj rad se bavi samo unaprijednim neuronskim mrežama (engl. *feedforward neural networks*). Cilj unaprijedne neuronske mreže je aproksimirati neku funkciju  $f^*$ , na način da definira preslikavanje  $y = f(\mathbf{x}; \boldsymbol{\theta})$  i nauči vrijednost parametara  $\boldsymbol{\theta}$  kako bi najbolje aproksimirala ciljanu funkciju  $f^*$ . Ovakvi modeli zovu se unaprijedni jer nema veza koje izlaze iz modela vraćaju nazad na ulaze.

### 2.1. Umjetni neuron

Warren McCulloch i Walter Pitts 1943. godine konstruirali su matematički model neurona kakav je prikazan na slici 2.1.



**Slika 2.1:** Umjetni neuron

Umjetni neuron prima više ulaza  $x_1, x_2, \dots, x_n$  koji tvore ulazni vektor  $\mathbf{x}$ . Taj ulazni vektor može biti stvarni ulaz ili izlaz iz nekog prethodnog neurona. Za svaku vrijednost

$x_i$  ulaznog vektora  $\mathbf{x}$  postoji vrijednost  $w_i$  koju nazivamo težina (eng. *weight*). To je vrijednost koja predstavlja utjecaj ulaza  $x_i$  na neuron. Svaki ulaz  $x_i$  množi se s odgovarajućom težinom  $w_i$  što daje umnožak  $x_i \cdot w_i$ . Vrijednosti  $w_1, w_2, \dots, w_n$  tvore vektor težina  $\mathbf{w}$ . Još se definira i vrijednost  $b$  koja označava pomak (engl. *bias*). Sve primljene vrijednosti se sumiraju prema izrazu 2.1.

$$net = \sum_{i=1}^n x_i \cdot w_i + b \quad (2.1)$$

Izraz 2.1 može se zapisati u matričnom obliku:

$$net = \mathbf{w}^\top \cdot \mathbf{x} + b \quad (2.2)$$

Pri tome ulazni vektor  $\mathbf{x}$  i vektor težina  $\mathbf{w}$  imaju dimenzije  $N \times 1$ , gdje  $N$  predstavlja broj ulaza u neuron, dok je pomak  $b$  skalar. Rezultat sumiranja  $net$  dalje se predaje kao ulaz prijenosnoj funkciji koja određuje konačni izlaz  $o$  neurona prema izrazu 2.3.

$$y = f(net) = f(\mathbf{w}^\top \cdot \mathbf{x} + b) \quad (2.3)$$

Gdje  $f$  predstavlja prijenosnu funkciju.

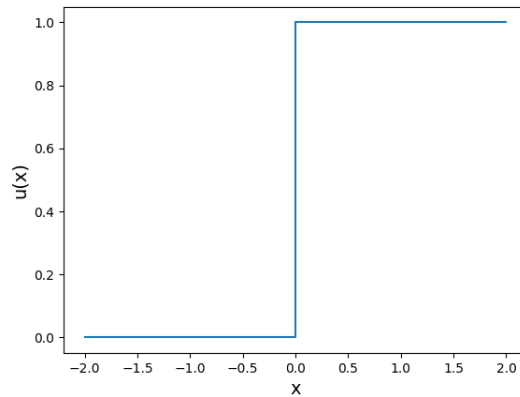
### 2.1.1. Prijenosne funkcije

Glavna zadaća prijenosne funkcije je pretvorba ulaznih vrijednosti u izlaznu vrijednost. Pri tome se mogu koristiti različite vrste funkcija ovisno o arhitekturi mreže. Danas postoji više prijenosnih funkcija koje se češće koriste.

Prva od njih je funkcija skoka i definirana je izrazom 2.4.

$$u(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (2.4)$$

Bitno svojstvo ove funkcije je prekid u točki  $x = 0$ , te zbog toga nije diferencijabilna u toj točki. Naime, to svojstvo onemogućava korištenje algoritama učenja koji se temelje na gradijentu. Izgled funkcije prikazan je na slici 2.2.

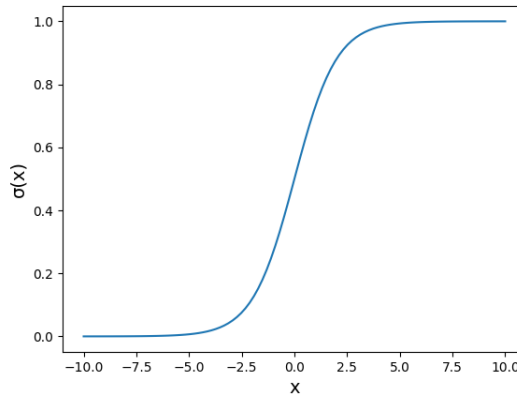


**Slika 2.2:** Funkcija skoka

Sigmoidalna funkcija ima jako dobra svojstva te se zbog toga često primjenjuje. Smješta bilo koji realni broj na interval  $[0, 1]$ . Definirana je izrazom 2.5.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

Ova funkcija ima dosta sličnosti s funkcijom skoka. Naime, ako je  $x$  neki veliki pozitivan broj onda je  $e^{-x} \approx 0$ , i izlaz iz neurona je blizu 1. S druge strane ako je  $x$  neki veliki negativni broj onda  $e^{-x} \rightarrow \infty$ , i izlaz iz neurona je blizu 0. To se jako dobro može primijetiti na grafu funkcije prikazanom na slici 2.3.



**Slika 2.3:** Sigmoidalna funkcija

Velika prednost sigmoidalne funkcije u odnosu na funkciju skoka je činjenica da je derivabilna, što omogućava korištenje algoritama učenja koji se temelje na gradijentu. Derivacija ove funkcije dana je izrazom 2.6.

$$\frac{d\sigma(x)}{dx} = \sigma(x) \cdot (1 - \sigma(x)) \quad (2.6)$$



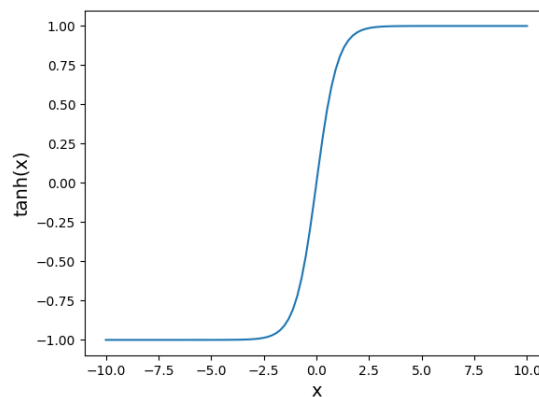
Na krajevima funkcija jako slabo reagira na promjene i zato će gradijent u tim dijelovima imati male vrijednosti. Ako mreža uči pomoću lokalnih gradijenata, u tom slučaju učenje staje, jer gradijenti ne mogu napraviti značajne promjene.

Funkciju tangens hiperbolni moguće je dobiti izravno iz sigmoidalne funkcije. Definicija je dana izrazom 2.7.

$$\tanh(x) = 2\sigma(2x) - 1 \quad (2.7)$$

Ova prijenosna funkcija smješta bilo koji realni broj na interval  $[-1, 1]$ . Prednost je što je derivabilna, ali kao i sigmoidalna funkcija na krajevima jako slabo reagira na promjene, te ako je vrijednost ulaza u funkciju u tom području, učenje staje, što se može vidjeti na grafu funkcije 2.4. Derivacija funkcije dana je izrazom 2.8.

$$\frac{d\tanh(x)}{dx} = 1 - \tanh^2(x) \quad (2.8)$$



**Slika 2.4:** Tangens hiperbolni

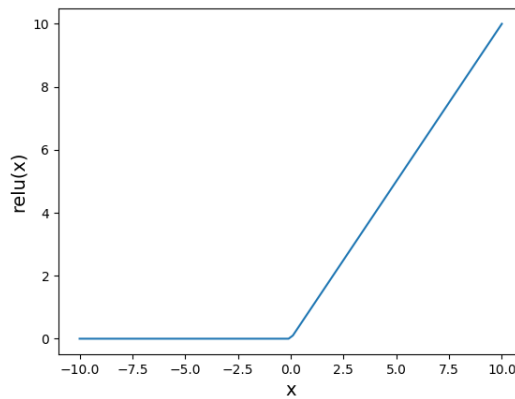
Zadnjih nekoliko godina sve se češće koristi ReLU (*Rectified Linear Unit*) prijenosna funkcija. Određena je izrazom 2.9.

$$\text{relu}(x) = \max(0, x) \quad (2.9)$$

Ova prijenosna funkcija sve vrijednosti veće od 0 propušta, dok vrijednosti manje od 0 uopće ne propušta, što se može vidjeti na grafu funkcije 2.9.

Derivacije ove funkcije dana je izrazom 2.10.

$$\frac{d\text{relu}(x)}{dx} = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2.10)$$

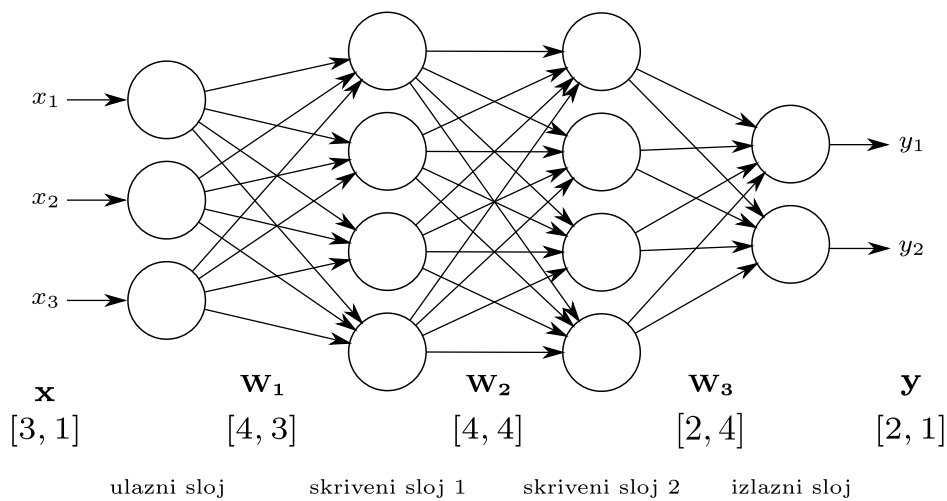


**Slika 2.5:** ReLU

Razlog zbog kojeg se sve češće primjenjuje je brzina računanja. Ali i ova prijenosna funkcija ima nedostatak. Iz izraza 2.10 se vidi da je za sve negativne brojeve derivacija jednaka 0, što predstavlja problem za algoritme učenja temeljene na gradijentu. Neuron kojemu je suma ulaza, odnosno *net* negativan, se zbog toga neće moći prilagođavati.

## 2.2. Arhitektura umjetne neuronske mreže

Umjetna neuronska mreža sastoji se od više umjetnih neurona koji su međusobno povezani i grupirani u slojeve. Na slici 2.6 je prikazana neuronska mreža koja ima četiri sloja. Prvi sloj je ulazni i sastoji se od tri neurona, a zadnji sloj je izlazni i sastoji se od dva neurona. Svaki sloj koji se nalazi između ulaznog i izlaznog zove se skriveni sloj.



**Slika 2.6:** Umjetna neuronska mreža

Ovakva mreža zove se još i potpuno povezana mreža, jer je izlaz iz svakog neurona u jednom sloju povezan sa svim neuronima u idućem sloju. Arhitektura neuronske mreže opisuje kompoziciju funkcija. Na primjer, mrežu sa slike 2.6 možemo opisati kompozicijom funkcija  $f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$ , gdje  $f^{(1)}$  predstavlja prvi skriveni sloj,  $f^{(2)}$  drugi skriveni sloj, a  $f^{(3)}$  izlazni sloj. Pri tome su funkcije  $f^{(1)}$ ,  $f^{(2)}$ ,  $f^{(3)}$  definirane izrazima:

$$\mathbf{h}_1 = f^{(1)}(\mathbf{x}; \mathbf{W}_1, \mathbf{b}_1) = g_1(\mathbf{W}_1 \cdot \mathbf{x} + \mathbf{b}_1) \quad (2.11)$$

$$\mathbf{h}_2 = f^{(2)}(\mathbf{h}_1; \mathbf{W}_2, \mathbf{b}_2) = g_2(\mathbf{W}_2 \cdot \mathbf{h}_1 + \mathbf{b}_2) \quad (2.12)$$

$$\mathbf{y} = f^{(3)}(\mathbf{h}_2; \mathbf{W}_3, \mathbf{b}_3) = g_3(\mathbf{W}_3 \cdot \mathbf{h}_2 + \mathbf{b}_3) \quad (2.13)$$

Gdje  $g_1$ ,  $g_2$  i  $g_3$  predstavljaju prijenosne funkcije u pojedinim slojevima, a matrice  $\mathbf{W}_1$ ,  $\mathbf{W}_2$  i  $\mathbf{W}_3$  predstavljaju matrice težina pojedinih slojeva. Vektori  $\mathbf{b}_1$ ,  $\mathbf{b}_2$  i  $\mathbf{b}_3$  predstavljaju pomak za svaki sloj.

## 2.3. Učenje umjetne neuronske mreže

Učenje neuronske mreže je proces nalaženja optimalnih parametara  $\theta$ , kako bi neuronska mreža što bolje aproksimirala ciljnu funkciju. Ciljna funkcija se zaključuje na osnovu uzoraka iz skupa za učenje koji se predočavaju neuronskoj mreži. Svaki uzorak je par koji se sastoji od ulaza u mrežu i željenog izlaza. Ovakav način učenja se zove nadzirano učenje (engl. *supervised learning*).

Kako bi odredili koliko dobro mreža aproksimira ciljnu funkciju koristimo funkciju gubitka. Minimiziranjem funkcije gubitka neuronska mreža se uči.

### 2.3.1. Funkcija gubitka

Kako bi se moglo ostvariti učenje, potrebno je definirati funkciju gubitka, koja ovisi o parametrima  $\theta$ , tj. o težinama  $\mathbf{W}$  i pomacima  $\mathbf{b}$ . Kao i prijenosna funkcija, funkcija gubitka se može definirati na više načina.

### **3. Konvolucijska neuronska mreža**

## **4. Zaključak**

Zaključak.

# LITERATURA

- [1] Ian Goodfellow, Yoshua Bengio, i Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [2] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, i Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *CoRR*, abs/1606.00373, 2016. URL <http://arxiv.org/abs/1606.00373>.
- [3] Marko Čupić. Umjetna inteligencija, umjetne neuronske mreže. <http://java.zemris.fer.hr/nastava/ui/ann/ann-20180604.pdf>.

## **Konvolucijski modeli za jednooku predikciju dubine scene**

### **Sažetak**

Sažetak na hrvatskom jeziku.

**Ključne riječi:** Ključne riječi, odvojene zarezima.

### **Title**

### **Abstract**

Abstract.

**Keywords:** Keywords.