

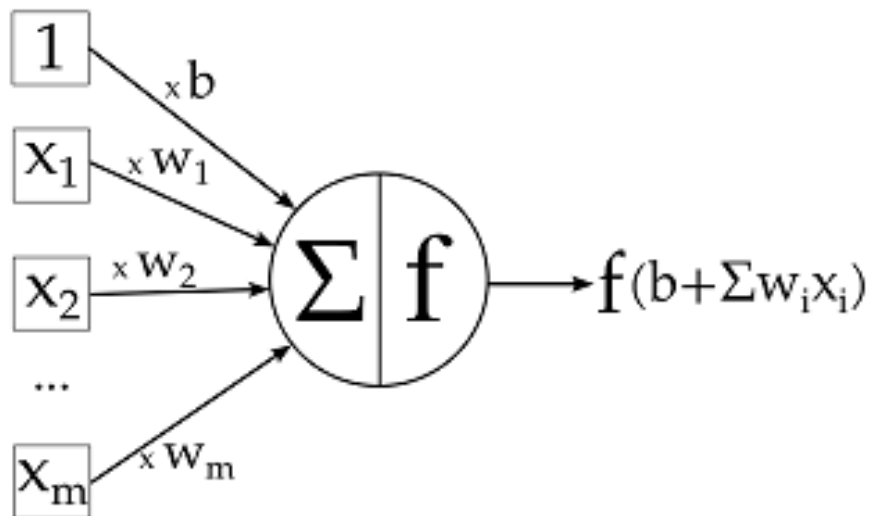
Oblast:

NEURONSKE MREŽE - REKAPITULACIJA

Veštački neuron

Model veštačkog neurona sastoji se od:

- **n ulaza**, gde je svaki ulaz skalar, odnosno realan broj
- **bias**, ulaz koji je zadat implicitno i čija vrednost je uvek 1.
- **n + 1 težina**, n za svaki ulaz + 1 za bias, težine su isto skalari, tj. realni brojevi
- **n + 1 množača**, za množenje svakog ulaza (i biasa) njemu odgovarajućom težinom
- **sabirač**, za sumu ponderisanih ulaza i biasa
- **aktivaciona funkcija**, čiji ulaz je rezultat sabirača, i uglavnom je ova funkcija neka nelinearna funkcija



Pošto ćemo veštački neuron implementirati kao graf izračunavanja, za početak potrebno je implementirati 3 čvora izračunavanja:

- množač
- sabirač
- sigmoidalnu funkciju (koja će biti aktivaciona funkcija)

Iako se graf i čvorovi izračunavanja koriste u i opštijem smislu, radi jednostavnosti implementacije uvešćemo par ograničenja za ova 3 čvora izračunavanja:

- množač: treba da ima **samo dva ulaza**, jer ćemo njega koristiti za množenje isključivo nekog konkretnog ulaza u neuron i njegove težine

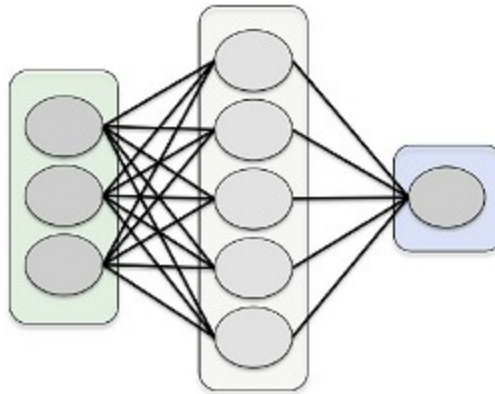
- sabirač: treba da ima **proizvoljan broj ulaza**, jer ćemo sumirati sve ponderisane ulaze u neuron, koliko god da ih ima
- sigmoidalna funkcija: treba da im **jedan ulaz**, jer ćemo je koristiti isključivo nad skalarom, odnosno rezultatom iz sabirača

Sloj veštačkih neurona

Više veštačkih neurona se mogu kombinovati u tzv. slojeve. Ograničenja prilikom pravljenja sloja neurona su:

- svi neuroni u sloju moraju imati isti broj ulaza
- svi neuroni u sloju moraju imati istu aktivacionu funkciju

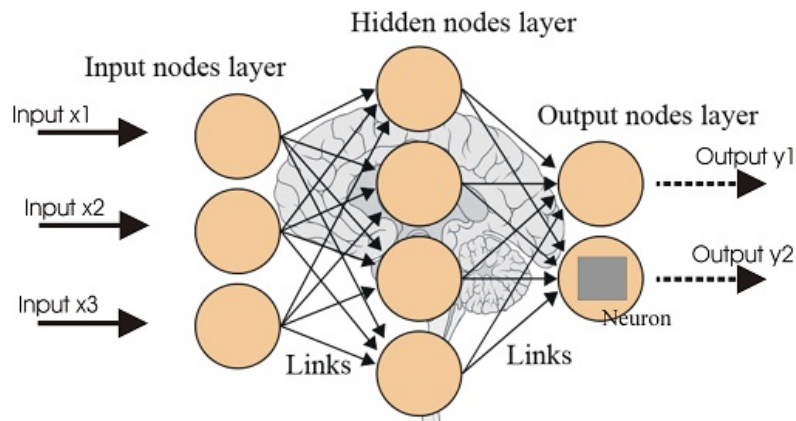
Sloj neurona ima onoliko izlaza koliko i ima neurona u sebi.



Veštačka neuronska mreža

Više slojeva neurona mogu se kombinovati u kompletnu veštačku neuronsku mrežu. Ukoliko mreža ima **N ulaza** i **M izlaza**, to znači da prvi sloj (ulazni sloj) neuronske mreže mora imati tačno N ulaza, i da poslednji sloj (izlazni sloj) mora imati M izlaza.

Neuronska mreža može imati proizvoljan broj skrivenih slojeva. Ulaz za svaki skriveni sloj je izlaz iz prethodnog sloja.



Obučavanje veštačke neuronske mreže

Za obučavanje neuronske mreže koristi se algoritam optimizacije prvog reda, odnosno algoritam opadajućeg gradijenta (*eng. gradient descent*). Kao funkciju greške koristimo kvadratnu grešku.

Kako bismo obučili neuronsku mrežu, neophodno je spremiti **ulaze i željene izlaze**, ovaj skup uređenih parova ulaza i željenih izlaza se naziva **obučavajući skup**.

Odnosno, ako neuronsku mrežu posmatramo kao crnu kutiju, ona treba da nauči funkciju $y=f(x_1, x_2, \dots, x_n)$.

Dakle, za zadate ulaze x , mreža treba da nauči kako da proizvede željeni izlaz y . Pa kako to neuronska mreža radi? Pa jednostavno -**optimizacijom težina svojih neurona!**

Formalno, treba minimizovati funkciju greške $E(W)$, gde W predstavlja **parametre** veštačke neuronske mreže, a ti **parametri su zapravo težine neurona**.

Dakle, za svaki uzorak iz svog obučavajućeg skupa:

1. izračunati izlaz iz neuronske mreže (forward-pass)
2. izračunati vrednost funkcije greške na osnovu izračunatog i željenog izlaza
3. izračunati gradijent funkcije greške u odnosu na izlaz
4. propagirati unazad (**backpropagation**) gradijent i računati gradijent težina svakog neurona (backward-pass)
5. ažurirati težine neurona korišćenjem algoritma opadajućeg gradijenta, odnosno za svaku

$$\text{težinu } w = w - \alpha * \frac{\delta E}{\delta w}$$

Gorenavedeni postupak iterativno ponavljati zadati broj iteracija/epoha (*nb_epochs*). Tokom obučavanja bi vrednost funkcije greške trebala da konvergira ka minimumu.

Zadatak:

Konstruisati i obučiti neuronsku mrežu na problemu određivanja da li se unutar prostorije nalazi neka osoba (Occupancy), ako su dati: vlažnost vazduha u prostoriji (Humidity), količina svetla u prostoriji (Light), količina CO2 u prostoriji (CO2), odnos težine vodene pare i ukupnog vazduha (HumidityRatio). Podaci za obučavanje se nalaze u `data/occupancy_train.csv`. Obučenu neuronsku mrežu validirati na podacima u `data/occupancy_test.csv`.