Poglavje 6

Nevronske mreže

V poglavju pregledamo arhitekturo in izpeljavo gradientov kriterijske funkcije za uteži najpreprostejšega model umetne nevronske mreže, tako imenovane usmerjene (angl. *feed forward*) nevronske mreže s polnimi povezavami med enotami sosednjih nivojev. Poleg strukture nevronske mreže uteži popolnoma določajo model nevronske mreže, izračun njihovih gradientov pa pri podanih učnih podatkih omogoča iskanje vrednosti uteži z gradientnim sestopom. Kot pri drugih modelih, ki smo jih pridobili na ta način, je tudi tu namen zgraditi mrežo, ki se čim bolj prilega učnim podatkom.

6.1 Arhitektura in notacija

Nevronska mreža je urejena mreža nevronov oziroma enot. Povezave med enotami označujejo funkcijske odvisnosti. Primer arhitekture mreže kaže slika 6.1. Vhod mreže sprejme atributno opisan primer oziroma vektor njegovih vrednosti atributov $\boldsymbol{x}$. Na izhodu mreža odda vrednosti izhodnih spremenljivk oziroma razredov. V strojnem učenju pravimo, da mreža za vhodni vektor $\boldsymbol{x}$ odda vektor napovedi $\boldsymbol{y}$. Vektor $\boldsymbol{y}$ je lahko eno ali več-dimenzionalen. Za regresijski model bo zadostovala ena enota, za klasifikacijo pa tipično izberemo toliko izhodnih enot, kolikor imamo razredov. Pri klasifikaciji enote poročajo o verjetnosti, da vhodnih primer $\boldsymbol{x}$ pripada določenemu razredu.

Vrednostim nevronov pravimo aktivacije. Aktivacije vhodnega nivoja mreže (L_1) nastavimo atributnim vrednostim primera, za katerega želimo izračunati vrednost razreda. Mreža z izračunom aktivacije enot njenem zadnjem nivoju (L_K) izračuna oceno vrednosti razrednih spremenljivk pripadajočega primera iz vhoda mreže. V preprosti nevronski mreži, ki jo obravnava to poglavje, so vse enote predhodnega nivoja povezane z vsemi enotami naslednjega nivoja.

Aktivacijo enote na nivoju L_i izračunamo kot uteženo vsoto vhodnih aktivacij nivoja L_{i-1}, ki jo transformiramo z neko nelinearno aktivacijsko funkcijo. Da bi linearne kombinacije vhodov lahko vključevale še konstantni člen, vsem nivojem nevronske mreže z izjemo zadnjega

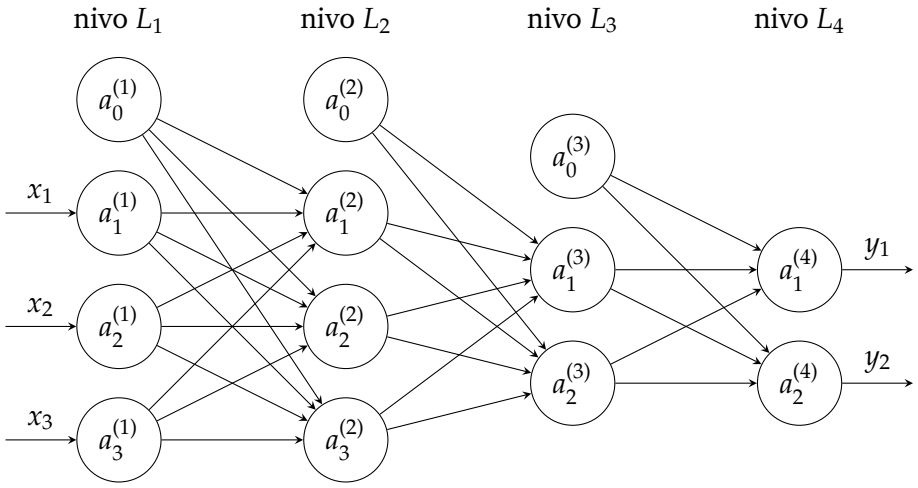
nivoja dodamo konstanten nevron $a_0^{(i)}$ ($i = \{1, 2, \ldots, K\}$), katerega aktivacija je enaka 1.

Aktivacijo j-tega nevrona na l-tem nivoju nevronske mreže izračunamo kot:

$$z_j^{(l)} = \sum_{k=0}^{n_{l-1}} w_{kj}^{(l)} \, a_k^{(l-1)} \tag{6.1}$$

$$a_j^{(l)} = \sigma(z_j^{(l)}), \tag{6.2}$$

kjer je n_k število enot nivoja L_k in σ aktivacijska funkcija. Dve tipični in pogosto uporabljani aktivacijski funkciji sta logistična funkcija, kjer je $\sigma(z) = 1/(1+\exp(-z))$ in popravljena linearna funkcija ali ReLU (angl. *rectified linear unit*), kjer je $\sigma(z) = \max(0, z)$. Aktivacijske funkcije morajo biti zvezne in odsekoma odvedljive.

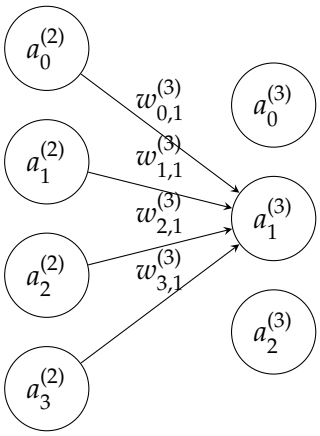


Slika 6.1: Primer štirinivojske nevronske mreže. Prvi nivo je nivo vhodnih podatkov, kjer aktivacijo nevronov nastavimo na vrednosti atributov ($a_i^{(1)} = x_i$). Sledita dva skrita nivoja, L_2 in L_3 in končni nivo, kjer so aktivacije nevronov vrednosti izračunanih ocen za razredne spremenljivke. V splošnem privzamemo, da imajo aktivacije $a_0^{(*)}$ na poljubnem nivoju konstantno vrednost 1.

V implementacijah nevronske mreže uteži na posameznem nivoju predstavimo z matrikami in aktivacije na posameznem nivoju z vektorji. Aktivacije $\boldsymbol{a}^{(l)}$ nivoja l tako izračunamo kot funkcijo skalarnega produkta uteži in aktivacij nivoja $l-1$:

$$\boldsymbol{a}^{(l)} = \sigma(\boldsymbol{a}^{(l-1)\mathsf{T}} \cdot W^{(l)}) \tag{6.3}$$

Primer: aktivacije drugega nivoja dela nevronske mreže s slike 6.2 naj zavzamejo vrednost



Slika 6.2: Primer sosednjih nivojev v nevronski mreži z oznakami uteži. Oznaka nivoja uteži ustreza oznaki nivoja, katerega aktivacijo računamo. Podobno je z vrstnim redom indeksov; prvi indeks je indeks nevrona, katerega vrednost je uporabljena pri računanju aktivacije, drugi pa indeks nevrona, za katerega aktivacijo računamo, drugi indeks pa indeks nevrona. Uteži, predstavljene na sliki, so del matrike uteži $W^{(3)}$ in predstavljajo njeno prvo vrstico.

$\boldsymbol{a}^{(2)} = (1\ 0\ 1\ 2)^\intercal$. Uteži tretjega nivoja predstavimo z matriko $W^{(l)}$, katere primer je:

$$W^{(l)} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Utežena vsota aktivacij, ki jo potrebujemo za izračun aktivacij tretjega nivoja nevronske mreže je tako:

$$\begin{pmatrix} 1 & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

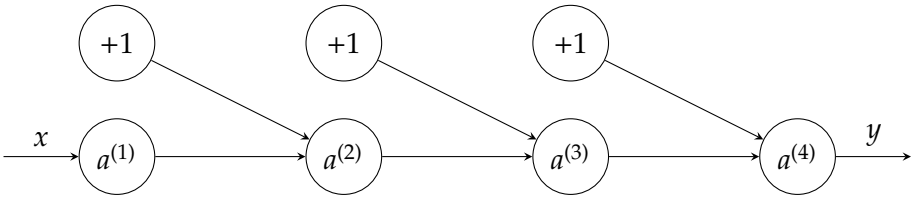
6.2 Učenje

Dan je nabor atributno opisanih primerov $\mathbf{X} \in \mathbb{R}^{m,n}$ s pripadajočo vrednostjo razreda. Nevronske mreže lahko obravnavajo primere z več razrednimi spremenljivkami, katerih vrednosti napovejo kot vrednost aktivacije enot v zadnjem nivoju nevronske mreže. Pri učenju želimo poiskati vrednosti uteži tako, da se napovedi mreže čimbolj ujemajo z dejanskimi vrednostmi razredov na učni množici primerov. Za učenje moramo tako določiti kriterijsko funkcijo, katere vrednost optimiziramo, in izračunati gradient kriterijske funkcije za vsako utež v mreži. Za optimizacijo vrednosti uteži uporabimo gradientni sestop. Slednjega že dobro poznamo, zato se bomo v tem poglavju ukvarjali samo z izračunom gradienta.

6.2.1 Enodimenzionalna nevronska mreža

Privzemimo, da imamo učno množico primerov, ki so opisani z enim samim atributom in kjer imamo en sam, zvezni razred. Najbolj enostavna nevronska mreža za tak primer vsebuje poleg enote s konstantno aktivacijo ((angl. *bias*)) eno samo aktivno aktivacijsko enoto na vsakem od nivojev. Primer take mreže je podan na sliki 6.2.1). Parametri te mreže so uteži $w_0^{(2)}$, $w_1^{(2)}$, $w_0^{(3)}$, $w_1^{(3)}$, $w_0^{(4)}$, $w_1^{(4)}$, kjer so z $w_0^{(*)}$ označene uteži, ki vodijo iz enot s konstantno aktivacijo in z $w_1^{(*)}$ označene uteži, ki vodijo iz enos s spremenljivo vrednostjo aktivacije. Za začetek si zamislimo, da imamo samo en učni primer (x, y). Kriterijska funkcija, ki jo optimiziramo oziroma za katero iščemo primerne parametre naj bo enaka kvadratu napake,

$$J\left(w_0^{(2)}, w_1^{(2)}, w_0^{(3)}, w_1^{(3)}, w_0^{(4)}, w_1^{(4)}\right) = \left(a^{(4)} - y\right)^2. \tag{6.4}$$



Slika 6.3: Nevronska mreža z enim samim aktivnim nevronom na vsakem od nivoju. Taka, poenostavljena, nam služi za pomoč pri izpeljavi gradientov uteži.

Iščemo gradiente kriterijske funkcije oziroma parcialne odvode kriterijske funkcije po vseh šestih utežeh. Najbolj enostavno nam je začeti čisto zadaj, pri zadnjem nivoju, ki ga označimo imenujmo nivo L. V našem primeru je $L = 4$. Odvisnost kriterijske funkcije od spremenljivk zadnjega nivoja so prikazane na sliki 6.4, izrazimo pa jih kot:

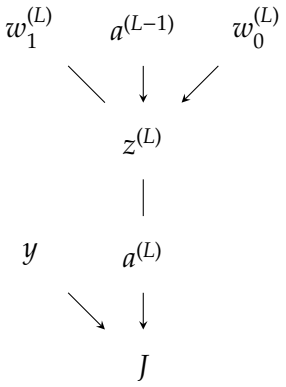
$$J = (a^{(L)} - y)^2, \tag{6.5}$$

$$a^{(L)} = \sigma(z^{(L)}), \tag{6.6}$$

$$z^{(L)} = w_1^{(L)} a^{(L-1)} + w_0^{(L)}. \tag{6.7}$$

Na tem, zadnjem nivoju, imamo dve uteži, $w_0^{(L)}$ in $w_1^{(L)}$. Izračunajmo gradient kriterijske funkcije po teh dveh utežeh. Najprej za utež $w_1^{(L)}$. Spomnimo so, da je kriterijska funkcija odvisna od aktivacije na nivoju L, ta od utežene vsote, in slednja od aktivacije nivoja $L-1$. Majhna sprememba uteži $w_1^{(L)}$ povzroči spremembo $z^{(L)}$, ta spremembo $a^{(L)}$, in slednja spremembo kriterijske funkcije J. Parcialni odvod kriterijske funkcije po uteži $w_1^{(L)}$ zato zračunamo z verižnim pravilom:

$$\frac{\partial J}{\partial w_1^{(L)}} = \frac{\partial z^{(L)}}{\partial w_1^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial J}{\partial a^{(L)}}, \tag{6.8}$$



Slika 6.4: Shematski prikaz odvisnosti kriterijske funkcije za zadnji nivo nevronske mreže.

kjer lahko naštete parcialne odvode dobimo iz zgoraj zapisanih funkcijskih odvisnosti:

$$\frac{\partial z^{(L)}}{\partial w_1^{(L)}} = a^{(L-1)}, \tag{6.9}$$

$$\frac{\partial a^{(L)}}{\partial z^{(L)}} = \sigma'(z^{(L)}), \tag{6.10}$$

$$\frac{\partial J}{\partial a^{(L)}} = 2(a^{(L)}) - y). \tag{6.11}$$

Tu je σ' odvod aktivacijske funkcije po njeni edini spremenljivki. V primeru logistične aktivacijske funkcije ta odvod že poznamo:

$$\sigma'(z^{(L)}) = \sigma(z^{(L)})(1 - \sigma(z^{(L)})) = a^{(L)}(1 - a^{(L)}) \tag{6.12}$$

Izračun parcialnega odvoda kriterijske funkcije po $w_1^{(L)}$ je bil enostaven. Kaj pa parcialni odvod po $w_0^{(L)}$? Dobimo ga enako kot zgoraj, drugačen je le odvod utežene vsote aktivacij na vhodu enote, ki je sedaj:

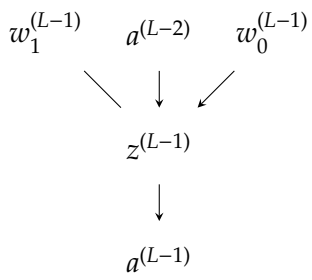
$$\frac{\partial z^{(L)}}{\partial w_0^{(L)}} = 1. \tag{6.13}$$

Do tu je šlo vse enostavno. Kaj pa predzadnji nivo. Tega (glej sliko 6.5) s kriterijsko funkcijo povezuje aktivacija enote $a^{(L-1)}$. Kako njena majhna sprememba učinkuje na spremembo kriterijske funkcije?

$$\frac{\partial J}{\partial a^{(L-1)}} = \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial J}{\partial a^{(L)}} \tag{6.14}$$

Zadnja dva parcialna odvoda smo že izračunali, manjka nam samo še prvi:

$$\frac{\partial z^{(L)}}{\partial a^{(L-1)}} = w_1^{(L)} \tag{6.15}$$



Slika 6.5: Shematski prikaz odvisnosti aktivacije zadnjega nivoja od elementov predzadnjega nivoja nevronske mreže.

Vse imamo pripravljeno za izračun gradientov za uteži na predzadnjem nivoju. Odvisnosti teh od aktivacije $a^{(L-1)}$ prikazuje slika 6.5, za to aktivacijo pa parcialni odvod že poznamo. Tudi tu parcialni odvod kriterijske funkcije po uteži $w_1^{(L-1)}$ dobimo z verižnim pravilom:

$$\frac{\partial J}{\partial w_1^{(L-1)}} = \frac{\partial z^{(L-1)}}{\partial w^{(L-1)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} \frac{\partial J}{\partial a^{(L-1)}} \tag{6.16}$$

Zadnji parcialni odvod že poznamo, izračunati moramo le še prva dva:

$$\frac{\partial z^{(L-1)}}{\partial w^{(L-1)}} = a^{(L-2)} \tag{6.17}$$

$$\frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} = \sigma'(z^{(L-1)}) = a^{(L-1)}(1 - a^{(L-1)}) \tag{6.18}$$

Podobno, z uporabo verižnega pravila, izpeljemo še gradiente za uteži preostalih nivojev ter pri tem uporabimo gradiente aktivacij, ki smo jih že izračunali. Izračun gradientov kriterijske funkcije po utežeh v nevronski mreži torej poteka od konca proti začetku. Postopku verižnega pravila v jeziku nevronskih mrež pravimo vzvratno razširanje napake oziroma (angl. *back propagation*).

Izpišimo torej splošne enačbe za izračun gradientov po tem postopku, in pri tem nek z l

označimo poljudni nivo nevronske mreže:

$$\frac{\partial J}{\partial w_1^{(l)}} = \frac{\partial z^{(l)}}{\partial w_1^{(l)}} \frac{\partial a^{(l)}}{\partial z^{(l)}} \frac{\partial J}{\partial a^{(l)}} \tag{6.19}$$

$$\frac{\partial J}{\partial a^{(l)}} = \begin{cases} 2(a^{(L)}) - y), & l = L \\ \frac{\partial z^{(l+1)}}{\partial a^{(l)}} \frac{\partial a^{(l+1)}}{\partial z^{(l+1)}} \frac{\partial J}{\partial a^{(l+1)}}, & l = 1, 2, \ldots, L-1 \end{cases} \tag{6.20}$$

$$\frac{\partial z^{(l)}}{\partial w_1^{(l)}} = a^{(l-1)} \tag{6.21}$$

$$\frac{\partial a^{(l)}}{\partial z^{(l)}} = \sigma'(z^{(l)}) \tag{6.22}$$

Podobno kot zgoraj lahko zapišemo tudi parcialne odvode za $w_0^{(l)}$, le da je tu parcialni odvod $\frac{\partial z^{(l)}}{\partial w_0^{(l)}}$ enak 1.

Zgoraj smo privzeli, da imamo samo en učni primer. Za množico primerov je kriterijska funkcija lahko:

$$J = \frac{2}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)^2 \tag{6.23}$$

kjer smo tokrat z $\hat{y}^{(i)}$ zapisali ocenjeno vrednost razredne spremenljivke za i-ti primer $x^{(i)}$ in z $y^{(i)}$ njeno pravo vrednost. Skladno s to kriterijsko funkcijo se pri izračunu gradientov spremeni le $\frac{\partial J}{\partial a^{(L)}}$, ki je sedaj enak povprečni vrednosti napake:

$$\frac{\partial J}{\partial a^{(L)}} = \frac{1}{m} \sum_{i=1}^{m} \left(a^{(L)}(x^{(i)}) - y^{(i)} \right), \tag{6.24}$$

kjer smo z $a^{(L)}(x^{(i)})$ označili vrednost aktivacije enote na zadnjem nivoju nevronske mreže, ki jo ta zavzame pri vhodnem primeru $x^{(i)}$.

Vsota kvadratov napake je samo ena od možnih kriterijskih funkcij. Pravzaprav lahko uporabimo katerokoli kriterijsko funkcijo, ki pa mora biti odvedljiva po aktivaciji zadnjega nevrona.

6.2.2 Splošna polno-povezana nevronska mreža

Zgornji primer "enodimenzionalne" nevronske mreže razširimo na primer, kjer imamo na vsakem nivoju lahko več aktivnih enot. Utežena vsota aktivacij na vhodu enote j nivoja l bo enaka:

$$z_j^{(l+1)} = \sum_{i=0}^{n_l} w_{ij} a_i^{(l)} \tag{6.25}$$

To pa je tudi edina sprememba, ki jo moramo upoštevati pri izračunu novih parcialnih odvodov. Odvod kriterijske funkcije po dani uteži ostane pravzaprav enak kot prej (z izjemo nekaj dodanih indeksov):

$$\frac{\partial J}{\partial w_{kj}^{(l)}} = \frac{\partial z_j^{(l)}}{\partial w_{kj}^{(l)}} \frac{\partial a_j^{(l)}}{\partial z_j^{(l)}} \frac{\partial J}{\partial a_j^{(l)}}, \tag{6.26}$$

spremeni se le odvod kriterijske funkcije po aktivaciji a_k^l, saj ta sedaj vpliva na kriterijsko funkcijo po različnih poteh, ki vodijo preko enot nivoja $l+1$. Tudi tu uporabimo verižno pravilo, a upoštevamo vse poti vplivov aktivacije a_k^l:

$$\frac{\partial J}{\partial a_k^{(l)}} = \sum_{j=0}^{n_l} \frac{\partial z_j^{(l+1)}}{\partial a_k^{(l)}} \frac{\partial a_j^{(l+1)}}{\partial z_j^{(l+1)}} \frac{\partial J}{\partial a_j^{(l+1)}} \tag{6.27}$$

Odvode izračunamo enako enostavno kot v "enodimenzionalni" mreži, le da imamo tu opravka z vsoto verig odvodov.

6.2.3 Vse skupaj, matrično in vektorsko

Zgornje izpeljave so zaradi preprostosti funkcij, ki gradijo umetne nevronske mreže, precej enostavne. A se hitro zapleteš z indeksi. Veliko preprosteje je vse te enačbe zapisati v vektorsko-matrični obliki. Pričnimo z učno množico $\mathbf{X} \in \mathbb{R}^{m,n}$ in za vse te naenkrat izračunajmo aktivacije enot nevronske mreže. Podatkom $\mathbf{X}$bomo dodali kolono enic, tako spremenjene podatke $\mathbf{X}'$ pa vložili v enote prvega nivoja nevronske mreže:

$$\underset{m\times n_1}{\mathbf{A}^{(1)}} = \underset{m\times(n+1)}{\mathbf{X}'} \tag{6.28}$$

Od tu dalje aktivacije izračunamo po naslednji enačbi:

$$\underset{m\times n_{l+1}}{\mathbf{Z}^{(l+1)}} = \underset{m\times n_l}{\mathbf{A}^{(l)}} \times \underset{n_l\times n_{l+1}}{\mathbf{W}^{(l+1)}}, \tag{6.29}$$

$$\underset{m\times n_{l+1}}{\mathbf{A}^{(l+1)}} = \sigma\Big(\underset{m\times n_{l+1}}{\mathbf{Z}^{(l+1)}}\Big) \tag{6.30}$$

kjer skalarno aktivacijsko funkcijo σ apliciramo na vsakega od elementov matrike v argumentu posebej.

S to, matrično, notacijo zapišimo tudi izračun gradientov. Pričnemo z gradientno matriko $\mathbf{D}^{(L)}$ za uteži zadnjega nivoja:

$$\underset{m\times n_L}{\mathbf{d}^{(L)}} = \Big(\underset{m\times n_L}{\mathbf{A}^{(L)}} - \underset{m\times n_L}{\mathbf{Y}}\Big) \circ \underset{m\times n_L}{\mathbf{A}^{(L)}} \circ \Big(1 - \underset{m\times n_L}{\mathbf{A}^{(L)}}\Big) \tag{6.31}$$

$$\underset{n_{L-1}\times n_L}{\mathbf{D}^{(L)}} = \frac{1}{m} \Big(\underset{n_{L-1}\times m}{\mathbf{A}^{(L-1)}}\Big)^{\mathsf{T}} \times \underset{m\times n_L}{\mathbf{d}^{(L)}} \tag{6.32}$$

kjer je $\circ$ oznaka za Hadamardov produkt, oziroma produkt po elementih dveh matrik. Nadaljujemo s izračunom gradientnih matrik $\mathbf{D}^{(L)}$ za uteži preostalih nivojev:

$$\underset{m\times n_l}{\mathbf{d}^{(l)}} = \left(\underset{m\times n_{l+1}}{\mathbf{d}^{(l+1)}} \times (\underset{n_{l+1}\times n_L}{\mathbf{W}^{(l+1)}})^\intercal\right) \circ \underset{m\times n_L}{\mathbf{A}^{(l)}} \circ \left(1 - \underset{m\times n_L}{\mathbf{A}^{(l)}}\right) \tag{6.33}$$

$$\underset{n_{l-1}\times n_l}{\mathbf{D}^{(l)}} = \frac{1}{m} \left(\underset{n_{l-1}\times m}{\mathbf{A}^{(l-1)}}\right)^\intercal \times \underset{m\times n_l}{\mathbf{d}^{(l)}} \tag{6.34}$$

V implementaciji nastavimo matrike uteži na neko manjšo, neničelno vrednost (na primer med 0 in 0.1) in potem z gradientnim sestopom izračunamo vrednost uteži, ki za dani nabor učnih podatkov minimizira kriterijsko funkcijo. Prileganju učnim podatkom se izogibamo z regularizacijo. Regularizacija L2 tudi tu prišteje kriterijski funkciji vsoto kvadratov uteži v nevronski mreži, s tem da se uteži na povezavah iz konstantnih enot mreže ne regularizira.