Domača naloga 5
Multinomska logistična regresija

Pravilnost implementacije

Pri implementaciji sem si pomagal z zapiski Softmax Regression[1]. Vse formule sem pretvoril v matrično obliko (edina `for` zanka je pri prečnem preverjanju).

Funkcija `softmax`

Poglejmo, da se pravilno obnaša na preprostem primeru.

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \theta = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Vrstice matrike X so primerki, stolpci pa atributi. Stolpci matrike θ pripadajo razredom, vrstice pa atributom.

Iz matrike θ se vidi, da prvi atribut prispeva k večji verjetnosti za 0. razred, drugi atribut pa k večji verjetnosti za 1. in 2. razred.

Rezultat funkcije `softmax` za prvo vrstico X bi moral imeti veliko verjetnost za 0. razred in maljhno za preostala dva. Rezultat za drugo vrstico pa bi moral biti: majhna verjetnost za 0. razred in enaka verjetnost za 1. in 2. rezred.

Funkcijo sem testiral na takem in podobnih primerih, in se obnaša po po pričakovanju opisanem zgoraj.

Funkcija `cost`

Ta funkcija bi morala vrniti vrednost blizu 0, če se napoved `softmax`-a ujema z oznakami y, sicer pa veliko negarivno vrednost.

Uporabimo isti primer kot prej in poženemo funkcijo `cost` z različnimi verdnosti za y in dobimo:

y	`softmax`
$\begin{bmatrix} 0 & 1 \end{bmatrix}$	-0.6932
$\begin{bmatrix} 0 & 2 \end{bmatrix}$	-0.6932
$\begin{bmatrix} 1 & 0 \end{bmatrix}$	-20.6933

Ti rezultati so smiselni, saj ima privi primerek največjo verjetnost za razred 0, drugi pa enoako verjetnost za razred 1 ali 2.

[1] `http://deeplearning.stanford.edu/tutorial/supervised/SoftmaxRegression/`

Funkcija `grad`

Spet uporabimo isti primer, tokrat za y izberemo [1 0] za katerega je θ najslabše prilagojena.

Gradient (zaokrožen) po posameznih elementih θ je:

$$\begin{bmatrix} -1 & 1 & 0 \\ 1 & -0.5 & -0.5 \end{bmatrix}$$

Za dani X in y bi morala biti θ čim bližje:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Če trenutni θ prištejemo gradient, bo bližje idealni vrednosti. Zato je rezultat smiselen.

Regularizacija

Pri povečevanju vpliva regularizacije (λ), vrednost `log loss` pri prečnem preverjanju na učnih podatkih počasi pada nato pa spet naraste, kar je v skladu s pričakovanji.

Opis reševanja napovedi

Podatke (`train.csv`, `test.csv`) sem s knjižnico `Pandas` in jim odstranil stolpec id. V tabeli učnih podatkov sem stolpec `target` spremenil v števila od 0 do 8. Nato sem podatke pretvoril v `numpy` sezname.

Na učnih podatkih sem izvedel prečno preverjanje ($k = 5$) in dobil naslednje rezultate:

λ	accuracy	log loss
0.0	0.762800	0.645148
0.1	0.762700	0.645068
0.2	0.762960	0.644904
0.4	0.762500	0.644552
0.8	0.762340	0.644525
1.6	0.762140	0.644292
3.2	0.762140	0.644163
6.4	0.761680	0.644251

Glede na te rezultate sem za regularizacijo uporabil $\lambda = 3.2$.

Za končno napoved sem model naučil na vseh učnih podatkih. Napoved modela sem shranil v `final.txt`.

Program za gradnjo končnega modela porabi **9 s**.