**1.Tulosta y\_train ennen vektori luokkien konvertointia binääriksiksi ja konvertoinnin jälkeen. Konvertoinnin jälkeinen tuloshan on konvoluutioverkon odotettu oikea tulos. Miksi tulos on juuri tuollainen?**

Tulos on hieman mielenkiintoinen konvertoinnin jälkeen. Tuo keras.utils.to.categorical muuttaa tässä tapauksessa y\_train:stä arvon 3 binääri matriisiin (esim jos muutettava arvo on 9 niin tulostukseen tulisi [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.] ja 0 niin [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]).

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti



**2. Selitä, miksi yksi suodatus tarkoittaa 10 säädettävää parametria. Ensimmäinen suodatus käsittää 32 suodatinta ja säädettäviä parametreja tuossa vaiheessa on 320 eli yhdessä suodattimessa pitää sitten olla 10 säädettävää parametria, mutta mitkä nuo parametrit ovat?**

10 säädettävää parametria koska 9 on satunnaista arvoa ja yksi bias arvo

**3. Selitä, miksi ensimmäisen suodatuksen Conv2D jälkeen kuvan koko pienenee 28\*28 kuvasta 26\*26 kuvaksi.**

Kuvan reunimmaisilla pikseleillä ei ole 8 pikseliä ympärillä, joten se poistaa reunimmaiset pikselit (eli ensimmäinen ja viimeinen sarake ja rivi pikseleitä poistetaan)

**4. Selitä, miksi kuvan koko puolittuu edellen MaxPooling 2D kuvassa.**

MaxPooling2D:ssä filterin koko 2x2 joka menee 26x26 kuvan päälle ja valitsee pikselin jolla on suurin arvo 2x2 filterissä, tämän takia kuvan koko puolittuu

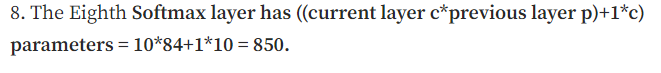
**5. Selitä, mistä tulevat toisen suodatuksen säädettävien parametrien luvut**

Text

Description automatically generated

Kuvassa näkyy miten parametrien määrä lasketaan. Eli tässä tapauksessa se lasketaan näin:   
((3 \* 3 \* 32) + 1) \* 64 = 18496

**6. Selitä, mistä tulee fully connected layerin säädettävien parametrien lukumäärät.**



Eli ((10\*1600)+1\*10) = 16010

**8. Tee koodi, jonka avulla voit löytää 60000 kuvan joukosta ne kuvat, joita CNN ei kykene tunnistamaan oikein.**

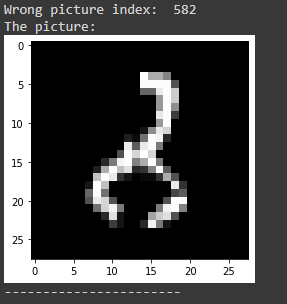
Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

Tämmöisellä koodilla löytyy ensimmäisestä 10000 kuvasta väärien kuvien indeksit (10000 kuvan läpikäymiseen kesti noin 6 min joten kaikkien 60000 kuvan läpikäynti kestäisi noin 30min). Koodi toimii siten että se ottaa mallin tuloksesta suurimman todennäköisyyden ja jos se on väärän numeron kohdalla niin koodi tulostaa kuvan indeksin.



Tein lopuksi tämmöisen if lauseen loppuun joka näyttää kuvan jos se oli väärin niin pääsee näkemään miltä se näytti. Pakko sanoa että joistakin kuvista en saa itsekkään selvää esim:



Kuva on kuulemma kahdeksan.