Zadanie 2

Načítanie dát

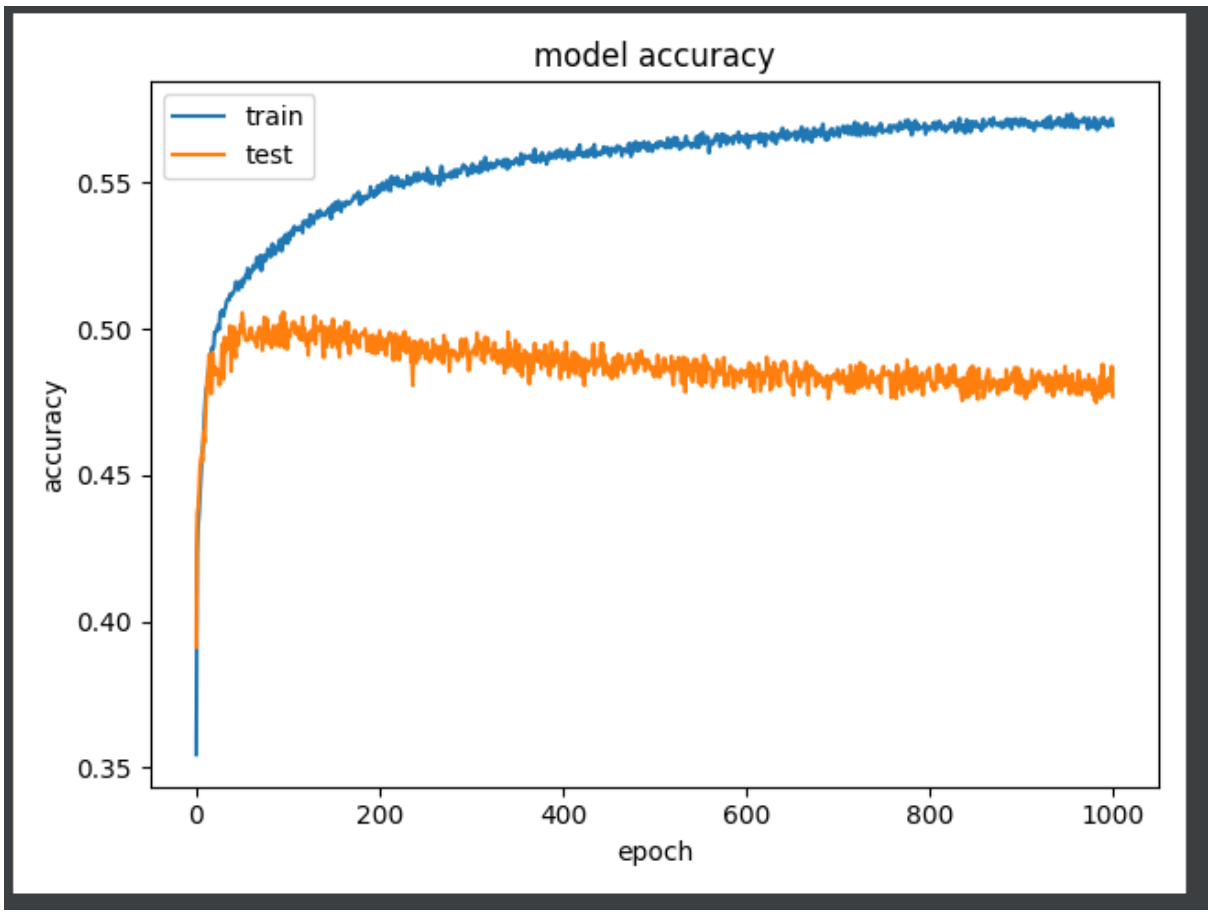
Dáta sa nachádzali v csv súbore. Načítal som ich pomocou knižnice pandas do takzvaného dataframu.

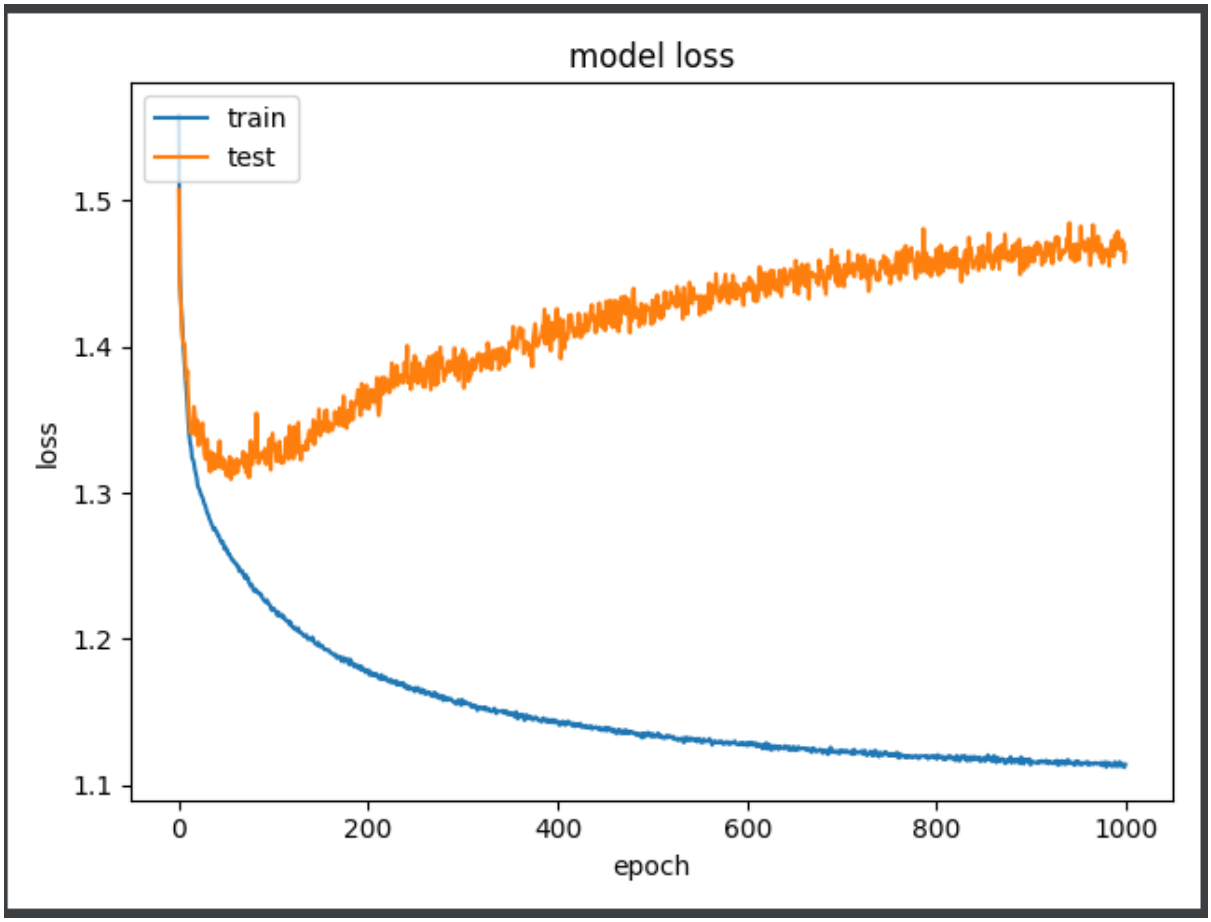
Predspracovanie dát

- Nájdenie miest bez hodnôt. Takýchto miest bolo minimum, preto som sa rozhodol pre ich odstránenie z datasetu.
- Odstránenie nepotrebných stĺpcov.
    - Stĺpce ako *track.id, track.name, track.album.id* a podobne nemajú žiadnu výpovednú hodnotu o pesničke ktorú reprezentujú. Odstránil som ich preto úplne z datasetu
- Nahradenie názvov žánru, číslami, a to nasledovne:
    - Edm - 0
    - latin – 1
    - pop – 2
    - R&b – 3
    - rap – 4
    - rock – 5
- Odstránenie hraničných hodnôt
    - Keďže veľká väčšina dát o pesničkách bola zhruba z rovnakej množiny dát, rozhodol som sa odstrániť príliš veľké/malé hodnoty. O potrebe vylúčiť jednotlivé hodnoty som sa rozhodoval na základe grafov.
- Nozmalizácia dát
    - Veľká väčšina nameraných hodnôt bola v rozmedzí od 0-1. Preto som sa rozhodol aj ostatné dáta normalizovať do tohto intervalu. Pri stĺpci *key* som dáta predelil číslom 11 (maximálna hodnota v danom stĺpci).
    - Pri stĺpcoch *duration, loudness a tempo* som využil naslednovný vzorec:
        - (aktuálnaHodnota - minimálnaHodnota)/(maximálnaHodnota - minimálnaHodnota)

Pretrénovanie

Jednou z našich úloh bolo pretrénovať neurónovú sieť. To nebol žiadny problém dosiahnuť. Pozrime sa na nasledovné dva grafy:


model accuracy
train
test
0.55
0.50
0.45
0.40
0.35
accuracy
0
200
400
600
800
1000
epoch


model loss
train
test
1.5
1.4
1.3
1.2
1.1
loss
0
200
400
600
800
1000
epoch

Na obidvoch grafoch jasne vidno známky pretrénovania siete. Jej úspešnosť a s pribúdajúcimi epochami stúpala, avšak iba na trénovacích dátach. Na testovacích to bolo presne opačne. To je spôsobené práve príliš veľkým zameraním na trénovacie dáta. Takéto pretrénovanie dosiahla sieť pri nasledujúcich parametroch:

- Tri vrstvy neurónov: 100,20,6
- Počet epoch: 1000

Riešenie pretrénovania

- Pomocou L2 regularizácie
  - Do skrytej vrstvy sme pridali túto regularizáciu

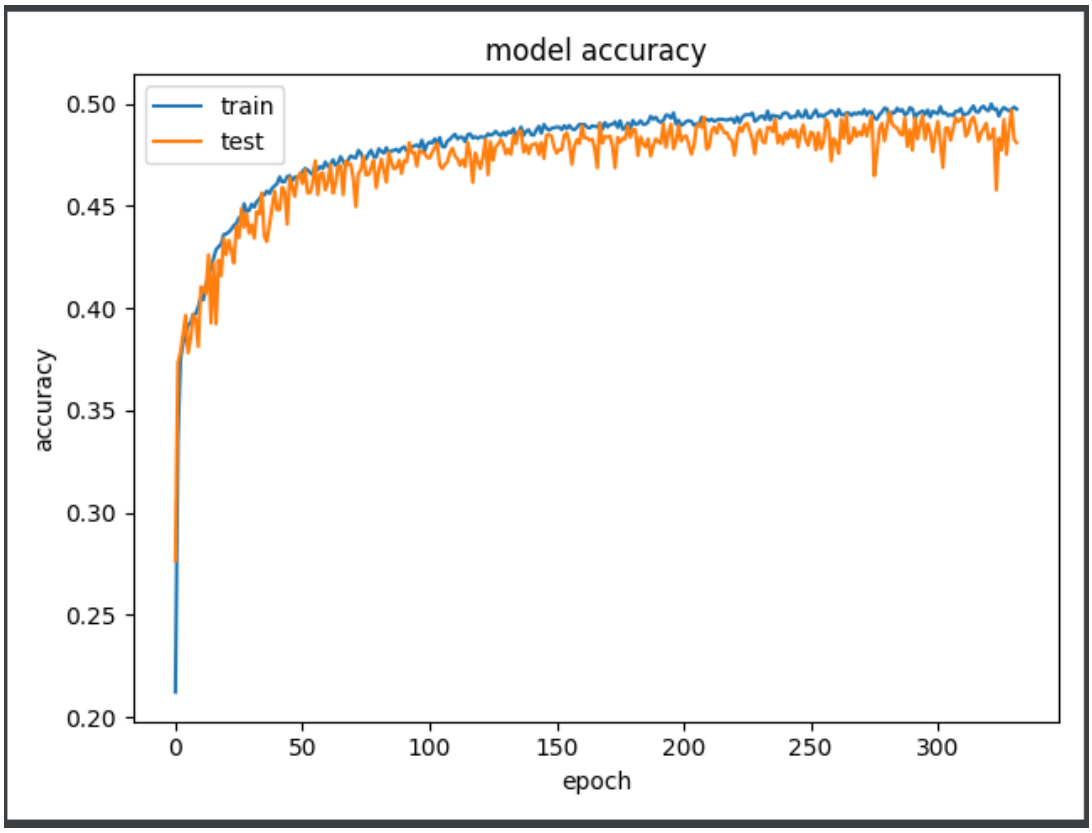
```
model = kr.Sequential()
model.add(kr.layers.Dense(100, input_dim=12, activation="sigmoid"))
model.add(kr.layers.Dense(20, kernel_regularizer=kr.regularizers.L2(0.01), activation="sigmoid"))
# model.add(kr.layers.Dense(20, activation="sigmoid"))
model.add(kr.layers.Dense(6, activation="sigmoid"))
```

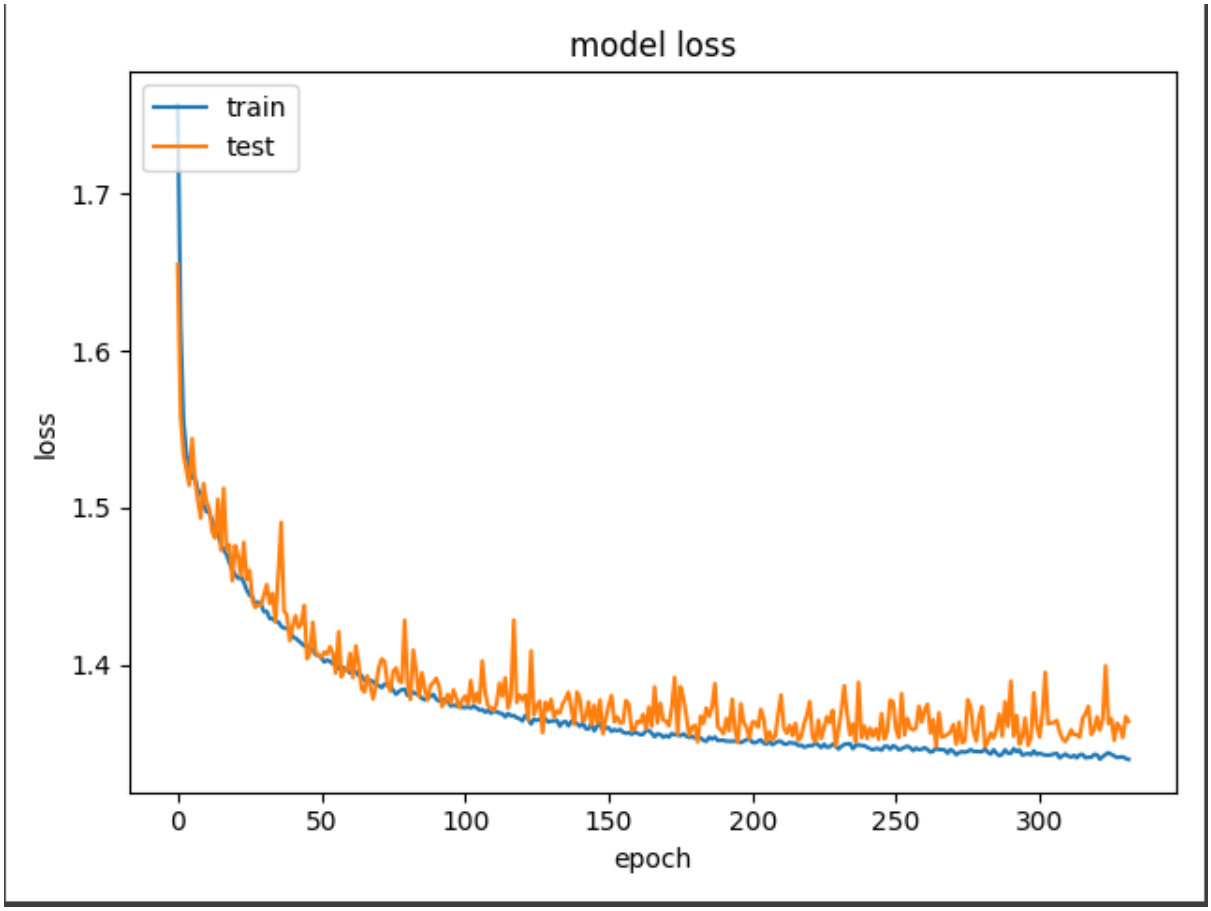
- Pomocou zastavovacej podmienky
  - Pre učenie neurónovej siete sme nastavili zastavovaciu podmienku. Ak sa validačná chyba nezmenši dostatočne v 50 po sebe idúcich epochách, zastavíme trénovanie.

```
early_stopping = kr.callbacks.EarlyStopping(monitor='val_loss', patience=50)

training = model.fit(train_x, train_y, epochs=1000, validation_data=(val_x, val_y), callbacks=[early_stopping])
```

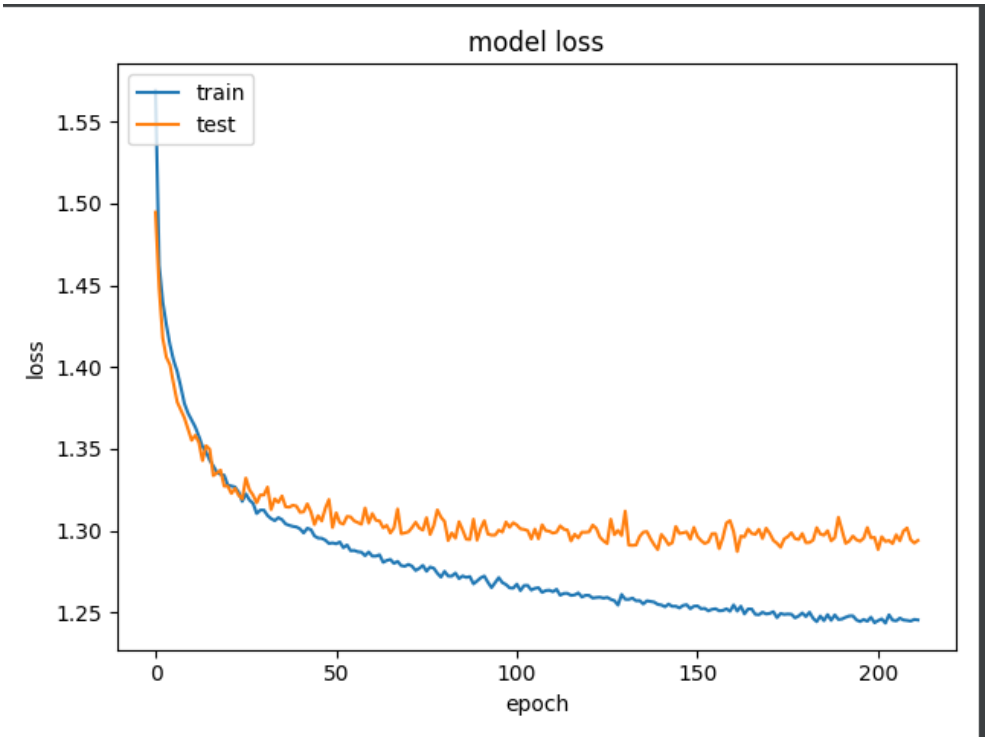
Na grafoch je možné vidieť aký to malo dopad na trénovanie:

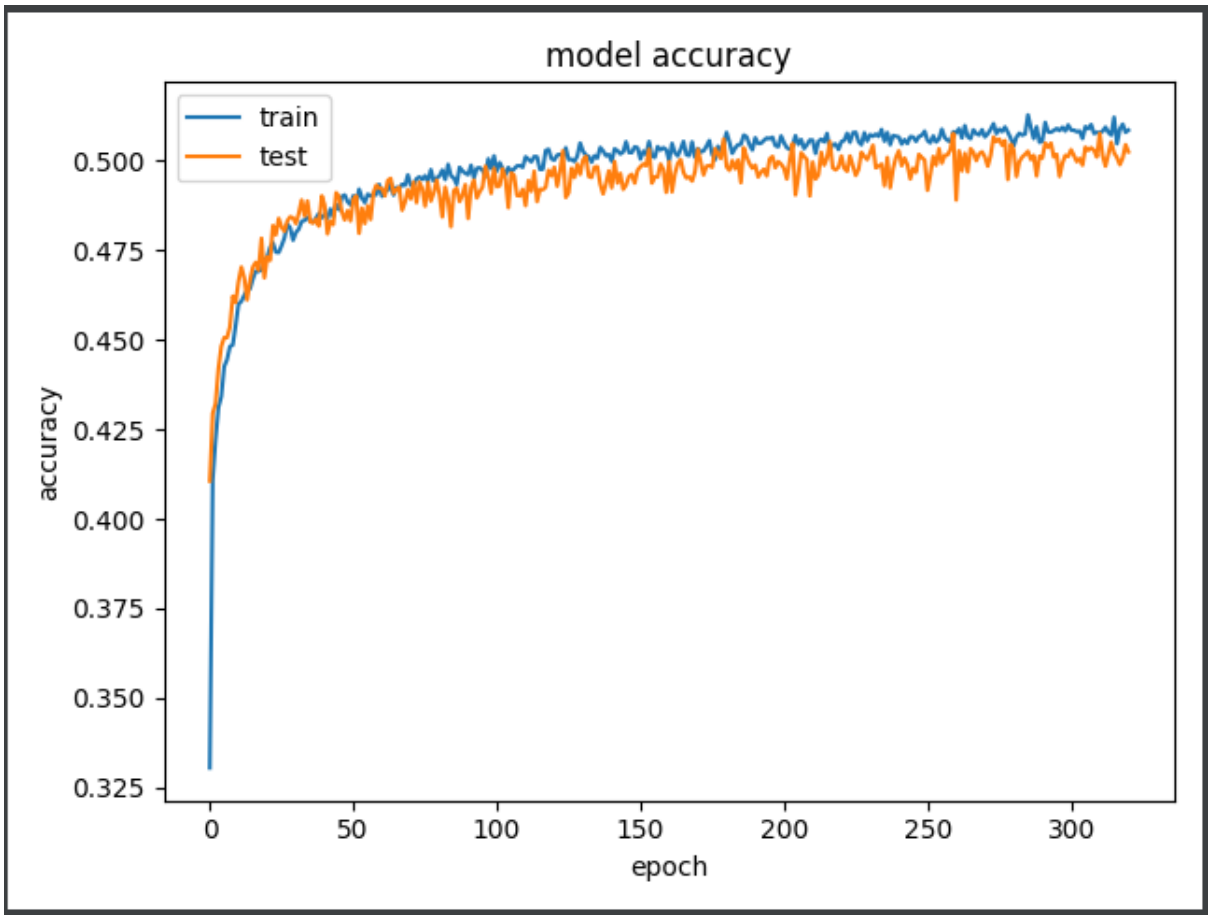




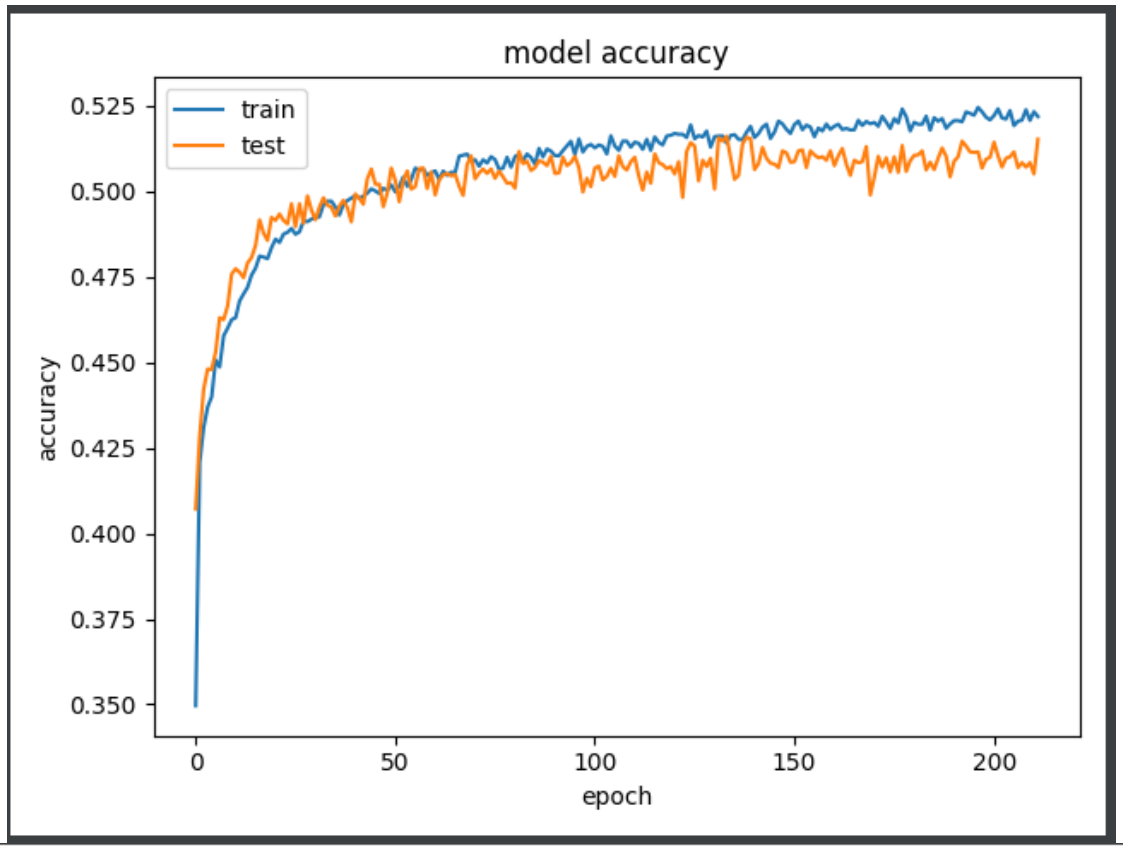
Môžeme si všimnúť že krivky sú viac „roztrasené“, to spôsobila L2 regularizácia. Taktiež je zjavné že sieť sa neučila celých 1000 epoch, ale skončila niekde okolo 330 epochy.

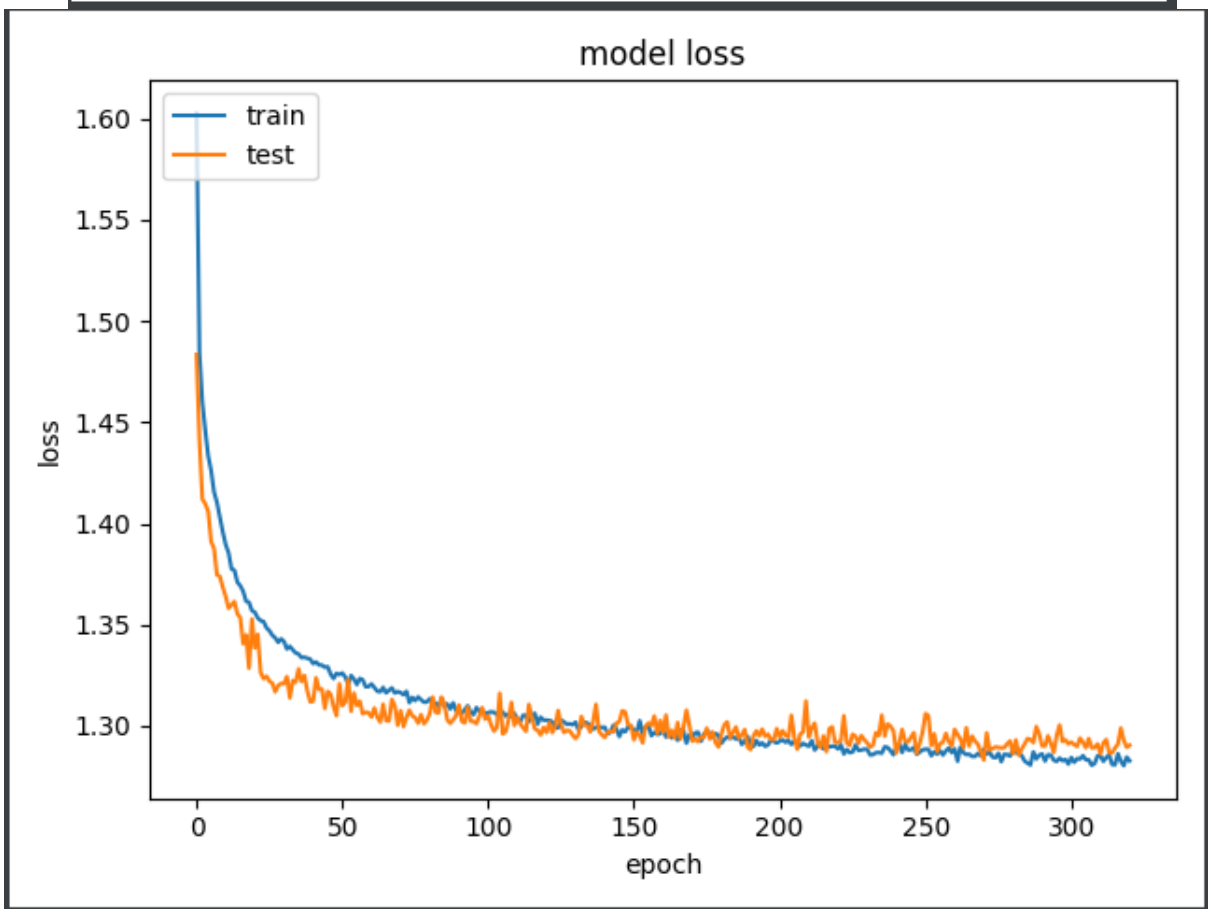
- Ďalšou možnosťou je dropout. Pri parametri droputu 0.2 sú výsledky siete nasledovné.




model accuracy
train
test
accuracy
epoch
0.500
0.475
0.450
0.425
0.400
0.375
0.350
0.325
0
50
100
150
200
250
300

- Dropout s parametrom 0.4

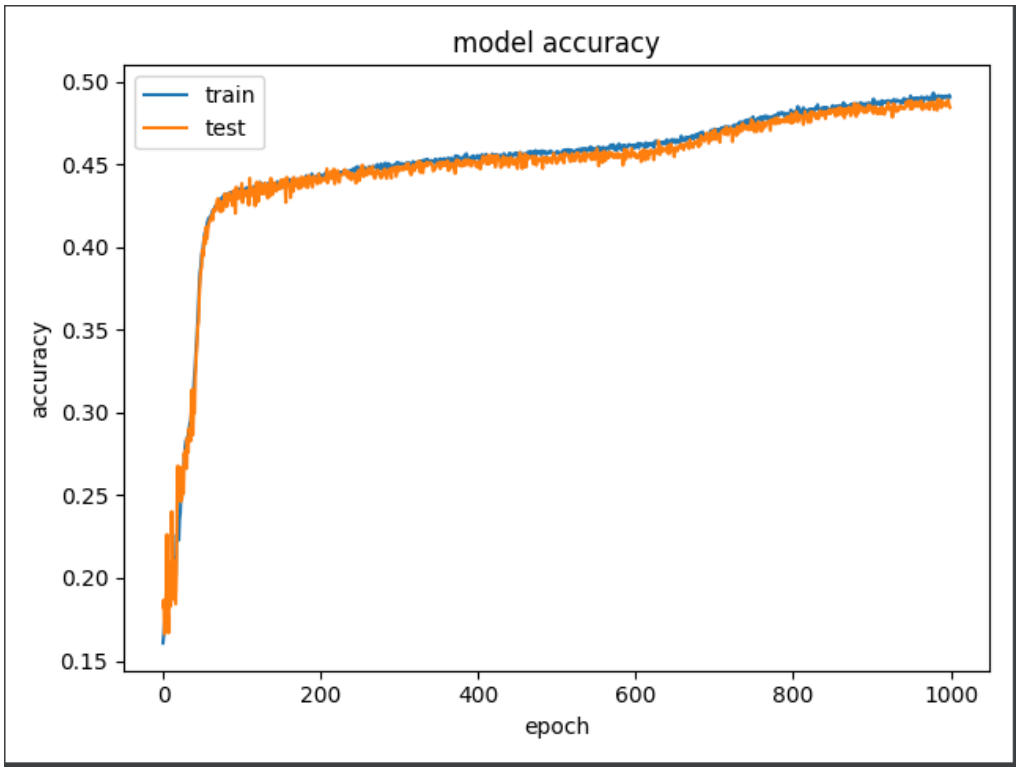


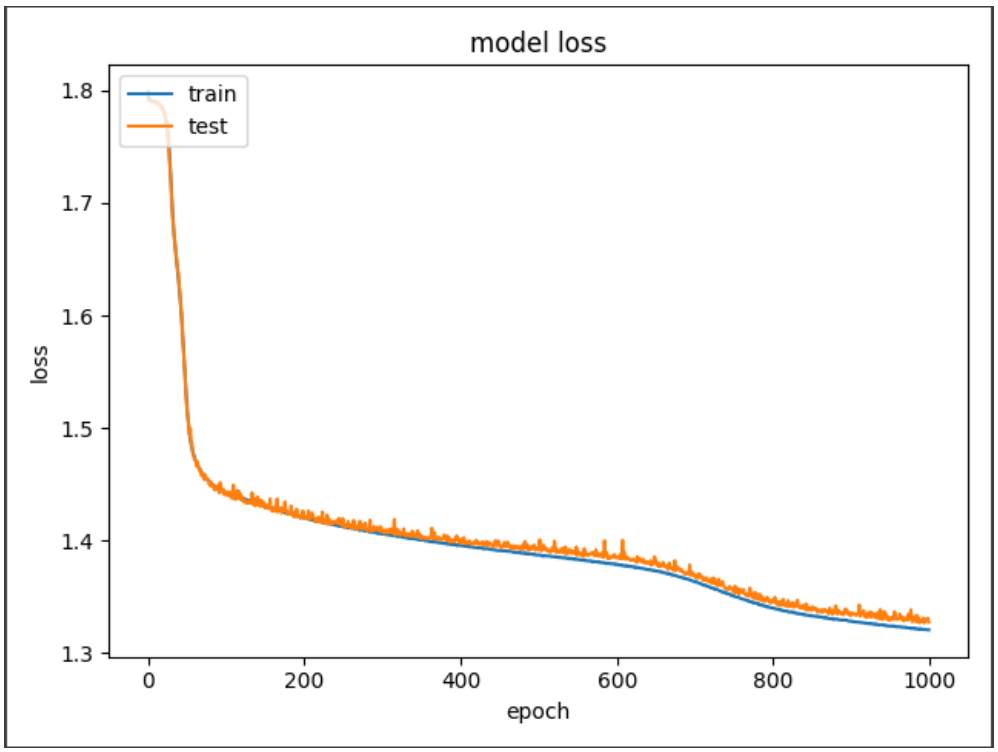


Vidíme že dropout s hodnotou 0.4 dosiahol o trochu lepšie lepšie výsledky. Hodnota dropoutu udáva koľko percent neurónov bude náhodne vynechaných.

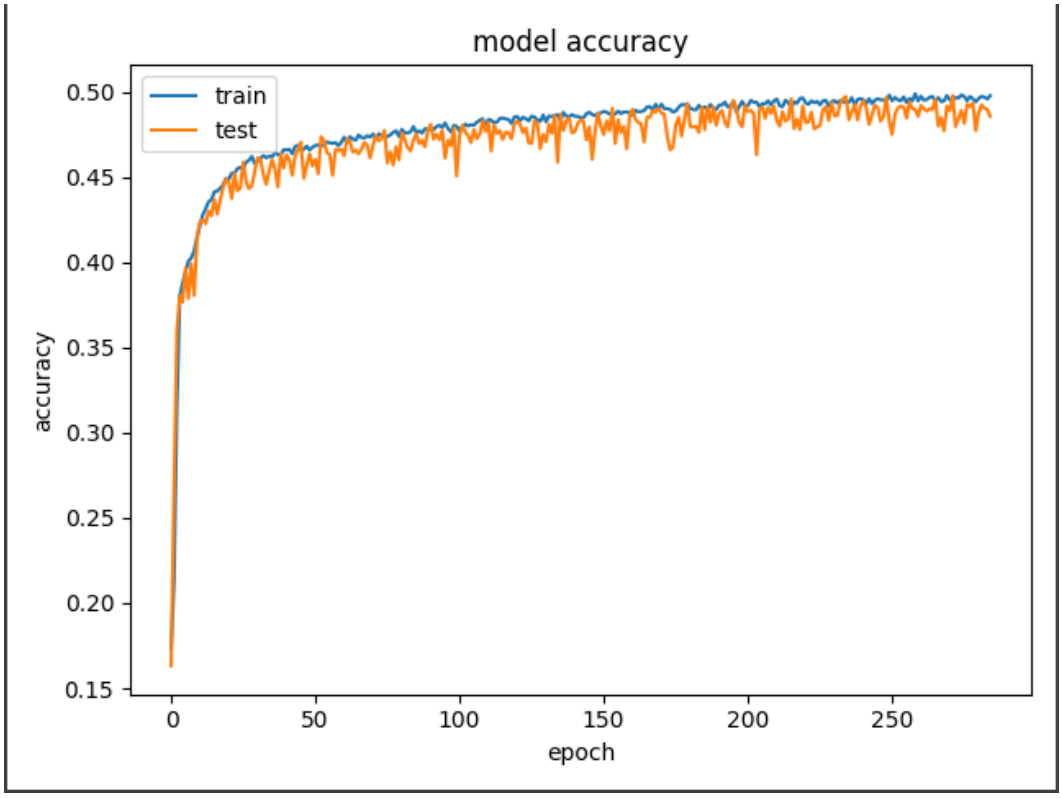
Nastavenia solverov

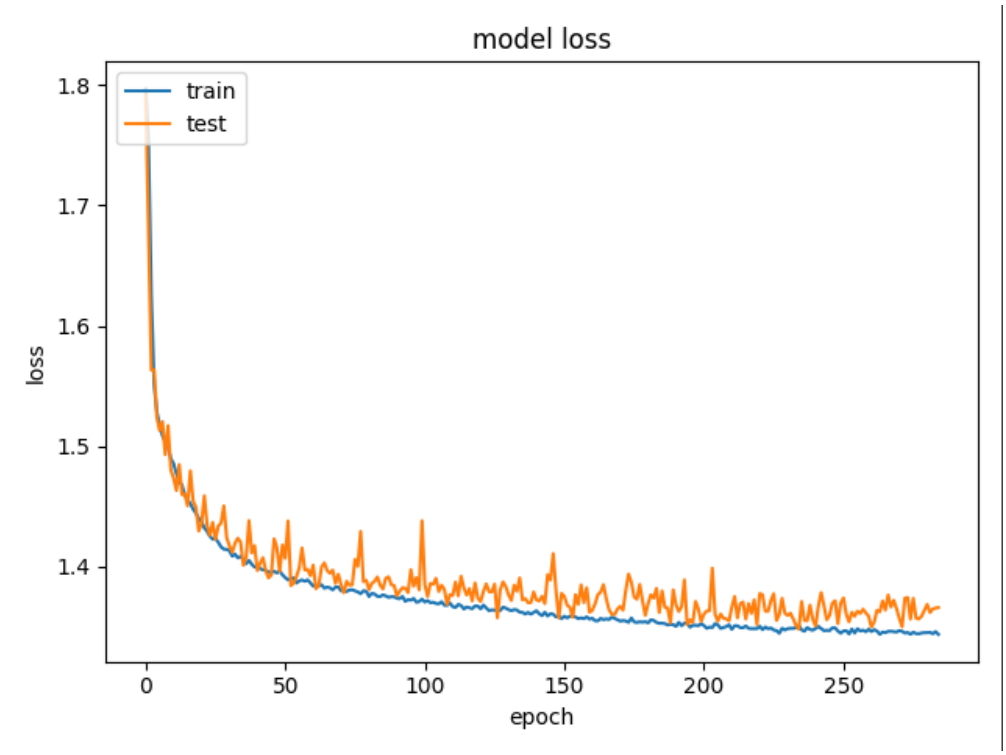
Budeme porovnávať solver adam so solverom sqg pri batchoch o veľkosti 20. Sgd dosiahol nasledujúce výsledky:





K solveru Adam som pridal ešte aj L2 regularizáciu, bez nej dochádzalo k značnému pretrénovaniu.





Môžeme vidieť že obidva solver-y mali približne rovnakú úspešnosť, avšak Adam sa ku konečnému výsledkzu dostal značne rýchlejšie. Už po necelých 300 epochách, zatiaľ či sgd na to potreboval až tisíc epoch.

SVM

Pri hľadaní ideálneho hyperparametra C som použil *GridSearchCV* z knižnice *sklearn*. Pri prvom pokuse som tam dal všetky dáta, to však trvalo neuveriteľne dlho. Ani po niekoľkých hodinách som sa nedočkal výsledku. Preto som toto hľadanie parametra C zopakoval s menším počtom dát (1000 riadkov) a z toho vyplynulo že najlepšie C sa rovná 2.5118864315095797.

Pri niekoľkých pokusoch takto natrénovať SVM bolo jej priemerné skóre (úspešnosť v percentách) cca 44%.

Úspešnosť najlepšej neurónovej siete bola okolo 50%. Preto moje konečné zhodnotenie je že neurónová sieť je o malý kúsok lepšia v predpovedaní žánra pesničiek ako SVM.