

# به نام خدا

## تمرین چهاردهم یادگیری عمیق

غزل زمانی نژاد

۹۷۵۲۲۱۶۶

1. سوال اول: جدول کلمات مشابه در تصویر زیر مشاهده می شود:

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10
cherry	(blossom, 0.65)	(peach, 0.61)	(plum, 0.59)	(berry, 0.55)	(blossoms, 0.54)	(pear, 0.54)	(orchard, 0.54)	(raspberry, 0.54)	(cherries, 0.54)	(mango, 0.53)
beach	(palm, 0.74)	(beaches, 0.67)	(fla., 0.66)	(florida, 0.57)	(resort, 0.57)	(shore, 0.55)	(pompano, 0.54)	(delray, 0.53)	(surf, 0.53)	(lauderdale, 0.52)
cake	(cakes, 0.78)	(chocolate, 0.72)	(cookies, 0.67)	(frosting, 0.66)	(dessert, 0.66)	(cookie, 0.65)	(pie, 0.64)	(baking, 0.63)	(bread, 0.63)	(pudding, 0.63)
jungle	(jungles, 0.75)	(lacandon, 0.54)	(mountainous, 0.52)	(hideout, 0.51)	(remote, 0.5)	(rugged, 0.49)	(forests, 0.48)	(countryside, 0.48)	(forest, 0.48)	(swampy, 0.47)
shoes	(shoe, 0.75)	(sneakers, 0.75)	(boots, 0.72)	(clothes, 0.7)	(socks, 0.69)	(clothing, 0.69)	(footwear, 0.68)	(leather, 0.68)	(jeans, 0.65)	(pants, 0.65)
success	(successful, 0.77)	(achieved, 0.76)	(successes, 0.76)	(winning, 0.64)	(thanks, 0.64)	(despite, 0.63)	(breakthrough, 0.63)	(best, 0.63)	(achieve, 0.63)	(enjoyed, 0.62)
airplane	(plane, 0.8)	(airplanes, 0.75)	(aircraft, 0.75)	(jet, 0.7)	(planes, 0.68)	(flight, 0.67)	(airliner, 0.66)	(jetliner, 0.66)	(flying, 0.64)	(crash, 0.6)
bird	(birds, 0.74)	(flu, 0.72)	(avian, 0.69)	(h5n1, 0.66)	(influenza, 0.62)	(virus, 0.6)	(migratory, 0.59)	(swine, 0.58)	(outbreaks, 0.58)	(chickens, 0.57)
music	(musical, 0.73)	(songs, 0.73)	(pop, 0.69)	(musicians, 0.69)	(recording, 0.68)	(jazz, 0.68)	(dance, 0.68)	(artists, 0.65)	(concert, 0.65)	(song, 0.65)
rain	(rains, 0.76)	(torrential, 0.71)	(downpour, 0.7)	(snow, 0.67)	(winds, 0.65)	(rainfall, 0.64)	(weather, 0.63)	(fog, 0.63)	(storms, 0.62)	(flooding, 0.62)

برای یافتن کلمات مشابه، فاصله میان بردار کلمه مورد نظر و سایر کلمات محاسبه شده است و کلماتی که

کمترین فاصله معنایی را داشته اند به عنوان مشابه ترین کلمات دیده می شوند. بررسی کلمات:

- برای هر کلمه مفرد/جمع، نزدیکترین کلمه به آن میتواند singular/plural آن باشد در صورتی که این مورد تنها برای cake، jungle، shoes، birds و rain دیده می شود.
- برای کلمه beach بسیاری از کلمات مشابه آن، نام شهرهای دارای ساحل هستند در صورتی که کلمات مرتبط تری در این دسته می تواند وجود داشته باشد.
- برای کلمه jungle، forest و forests مترادف آن هستند اما در جایگاه 9 و 7 قرار دارند.
- برای کلمه success کلمه ششم معنای "علیرغم" دارد در صورتی که به موفقیت ربط پیدا نمیکند.
- برای کلمه bird، بسیاری از کلمات بیماری های مربوط به پرندگان است، در صورتی که نام گونه های خاص پرنده و یا حیوانات ربط بیشتری به آن پیدا می کند. نشان می دهد context ای که در آن embedding ها آموخته شده، به بیماری های پرندگان پرداخته است.
- برای کلمه music، کلمه song مترادف آن است در صورتی که در جایگاه دهم قرار دارد.

## سوال دوم:

	Near Word	Far Word
boy	(man, 0.32)	(palace, 0.85)
bus	(road, 0.43)	(jungle, 0.82)
bread	(flour, 0.34)	(snow, 0.78)
thin	(slim, 0.57)	(truck, 0.91)
sun	(sunny, 0.54)	(jam, 0.86)

در اینجا درصد دور بودن کلمات far words بسیار بیشتر از درصد نزدیک بودن کلمات near words است و درصد تشابه کلمات نزدیک مطابق انتظار نبود. مثلا کلمه boy و man که هردو به جنس مذکر اشاره دارند، تنها 32 درصد شباهت داشته اند. علت آن به context ای که در آن embedding ها آموخته شده برمی گردد. به همین دلیل عملکرد امبدینگ های پیش آموخته در تسک های مختلف، به تسک مربوطه و محتوایی که امبدینگ در آن train شده مربوط است.

## سوال سوم: استفاده از first\_distances:

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10
('king', 'woman', 'man')	(queen, 0.7)	(princess, 0.61)	(monarch, 0.59)	(throne, 0.58)	(prince, 0.58)	(elizabeth, 0.55)	(daughter, 0.54)	(kingdom, 0.53)	(mother, 0.52)	(crown, 0.52)
('actor', 'girl', 'boy')	(actress, 0.87)	(starring, 0.71)	(actresses, 0.69)	(actors, 0.69)	(starred, 0.68)	(screenwriter, 0.63)	(comedian, 0.63)	(film, 0.61)	(movie, 0.6)	(filmmaker, 0.58)
('doctor', 'she', 'he')	(nurse, 0.7)	(mother, 0.6)	(woman, 0.6)	(her, 0.59)	(physician, 0.57)	(pregnant, 0.57)	(dr., 0.56)	(doctors, 0.56)	(patient, 0.55)	(hospital, 0.55)
('homemaker', 'she', 'he')	(housewife, 0.71)	(schoolteacher, 0.61)	(widowed, 0.55)	(businesswoman, 0.55)	(mom, 0.55)	(waitress, 0.53)	(hairstylist, 0.53)	(mother, 0.52)	(socialite, 0.52)	(grandmother, 0.51)
('football', 'woman', 'man')	(basketball, 0.67)	(soccer, 0.64)	(volleyball, 0.58)	(league, 0.55)	(softball, 0.55)	(hockey, 0.54)	(rugby, 0.53)	(ncaa, 0.52)	(club, 0.52)	(collegiate, 0.52)

## استفاده از second\_distances:

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10
('king', 'woman', 'man')	(prince, 0.55)	(ii, 0.54)	(brother, 0.54)	(iii, 0.53)	(reign, 0.53)	(uncle, 0.52)	(kingdom, 0.5)	(henry, 0.5)	(kings, 0.5)	(iv, 0.48)
('actor', 'girl', 'boy')	(comedian, 0.61)	(starring, 0.6)	(actors, 0.58)	(starred, 0.56)	(movie, 0.55)	(brother, 0.55)	(father, 0.55)	(film, 0.53)	(musician, 0.53)	(filmmaker, 0.52)
('doctor', 'she', 'he')	(physician, 0.66)	(surgeon, 0.57)	(doctors, 0.57)	(medical, 0.56)	(him, 0.54)	(dr., 0.54)	(himself, 0.53)	(his, 0.52)	(hospital, 0.52)	(man, 0.51)
('homemaker', 'she', 'he')	(43-year, 0.59)	(schoolteacher, 0.59)	(42-year, 0.55)	(housewife, 0.55)	(55-year, 0.54)	(48-year, 0.53)	(bricklayer, 0.53)	(47-year, 0.52)	(44-year, 0.52)	(39-year, 0.52)
('football', 'woman', 'man')	(soccer, 0.68)	(baseball, 0.64)	(team, 0.63)	(basketball, 0.62)	(league, 0.62)	(players, 0.61)	(rugby, 0.61)	(club, 0.6)	(game, 0.58)	(hockey, 0.58)

در اینجا باید دقت کنیم دچار بایاس نسبت به مواردی از جمله جنسیت، نژاد و .. نشویم.

استفاده از فاصله دوم بهتر است، زیرا با بایاس مقابله کرده است. مثلا در جدول اول، نسبت دکتر و جنس مذکر به نسبت پرستار و جنس مونث مپ شده است. که در اینجا درجه این دو شغل باهم متفاوت است. اما در جدول دوم همان نسبت به physician (کلمه مترادف doctor) مپ شده است. یا در جدول اول نسبت فوتبال و مرد به

نسبت بسکتبال و زن مپ شده است. در صورتی که ورزش ها مختص جنسیت خاصی نیستند. در جدول دوم همان نسبت به کلمه soccer (مترادف فوتبال) مپ شده است.

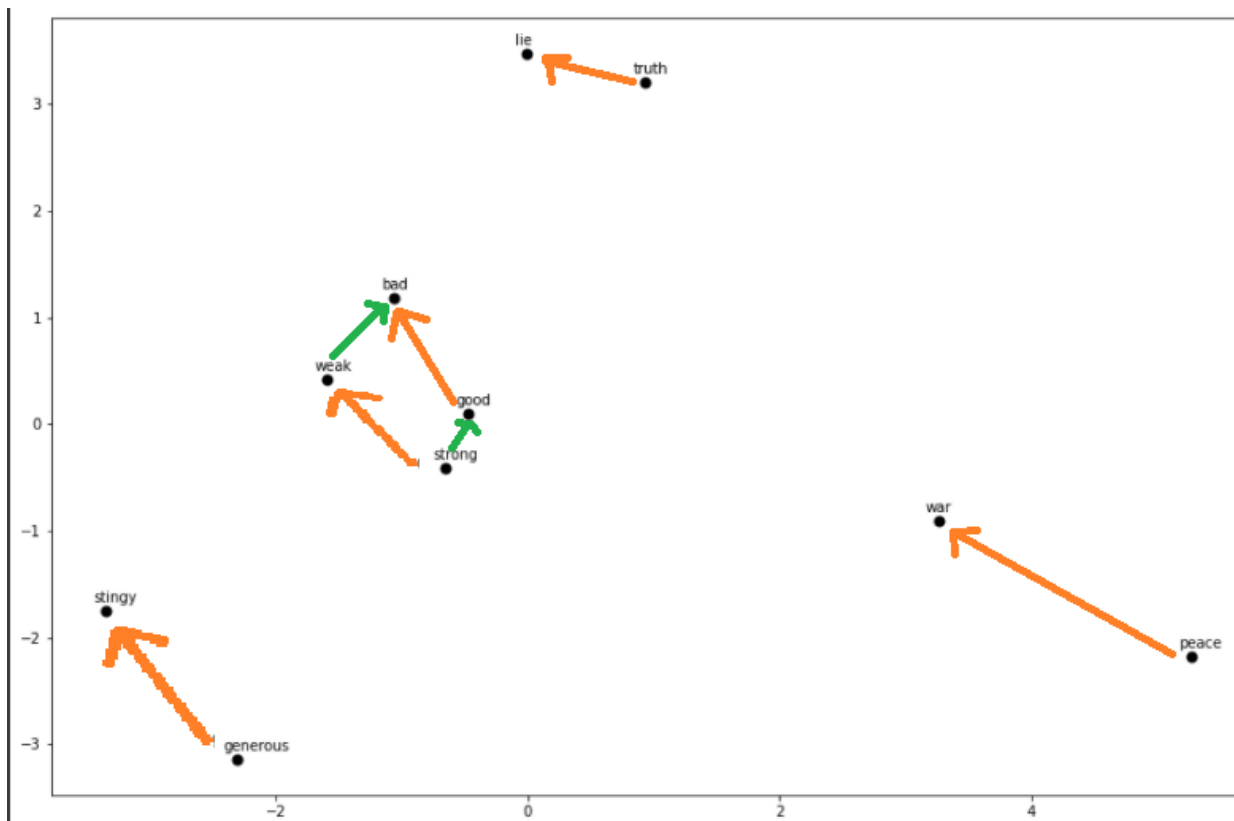
#### سوال چهارم:



مطابق تصویر بردار میان  $actor \rightarrow actress$  و  $boy \rightarrow girl$ ,  $man \rightarrow woman$ ,  $king \rightarrow queen$  یک جهت و یک سو هستند که این نشان دهنده رابطه معنایی میان کلمات است. دو سر هر بردار نشان دهنده یک کلمه اما با جنسیت های مختلف است.

از طرفی بردار میان  $girl \rightarrow woman$  و  $boy \rightarrow man$  نیز مشابه هستند (می توانستیم این بردارها را مثلاً برای  $girl \rightarrow queen$  و  $boy \rightarrow king$  نیز رسم کنیم).

### سوال پنجم:

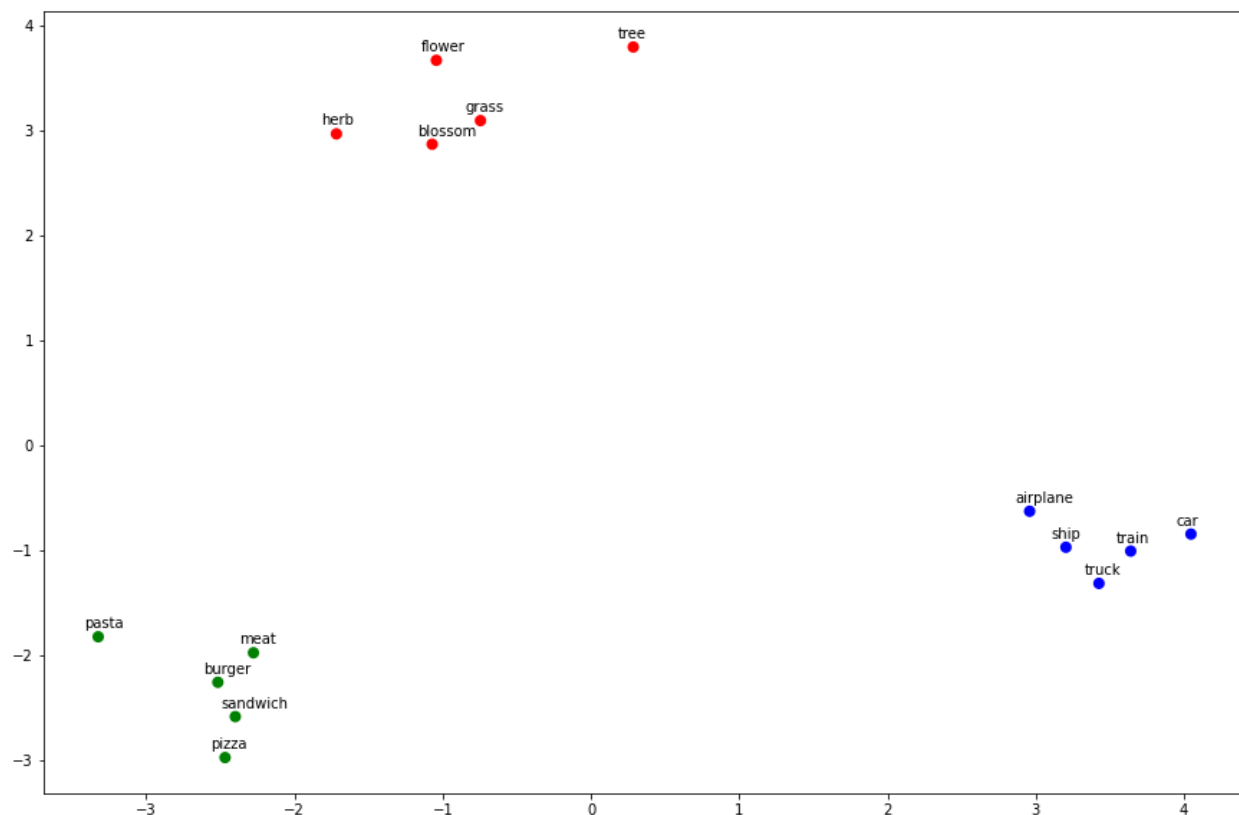


مطابق تصویر بردار میان  $\text{truth} \rightarrow \text{lie}$ ,  $\text{peace} \rightarrow \text{war}$ ,  $\text{strong} \rightarrow \text{weak}$ ,  $\text{generous} \rightarrow \text{stingy}$  و  $\text{good} \rightarrow \text{bad}$  همگی در یک جهت و یک سو هستند که این نشان دهنده رابطه معنایی میان کلمات است. کلمات دو سر هر بردار متضاد هستند.

از طرفی بردار میان  $\text{strong} \rightarrow \text{good}$  و  $\text{weak} \rightarrow \text{bad}$  نیز مشابه هستند (می توانستیم این بردارها را مثلاً برای  $\text{stingy} \rightarrow \text{bad}$  و  $\text{generous} \rightarrow \text{good}$  و .. نیز رسم کنیم). در اینجا بردار میان کلمات مثبت به  $\text{good}$  و کلمات منفی به  $\text{bad}$  است.

**سوال ششم:** در اینجا کلماتی که از نظر معنایی در یک دسته قرار می گیرند، در صفحه مختصات نیز به نوعی در یک cluster واقع شده اند. همچنین در گروه حیوانات، کلمه lion که نسبت به بقیه حیوانات وحشی است، فاصله برداری دورتری از سایر حیوانات دارد. در گروه کشورها، ایران در فاصله بیشتری نسبت به کشورهای عربی قرار دارد.

### سوال هفتم:



همانطور که انتظار میرفت، کلماتی که از یک گروه هستند در یک cluster قرار گرفته اند. در گروه غذاها، pasta که نسبت به بقیه متفاوت تر است، دورتر قرار گرفته. در گروه وسایل نقلیه، ماشین، قطار و کامیون که حمل و نقل زمینی هستند به یکدیگر نزدیکترند. در گروه گیاهان، گل، علف و شکوفه که معنای نزدیکی دارند به هم نزدیکترند.

### سوال هشتم:

```
3 word = 'calf'
4 similar_words = dict()
5 similar_words[word] = sorted(wv_from_bin.most_similar(word), key=lambda item: item[1], reverse=True)
6
7 show_DataFrame(similar_words, [f'Word {i + 1}' for i in range(10)])
```

Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10	
calf	(thigh, 0.74)	(groin, 0.71)	(hamstring, 0.68)	(ankle, 0.67)	(knee, 0.66)	(tendon, 0.62)	(calves, 0.62)	(sprained, 0.61)	(bruised, 0.58)	(rib, 0.58)

در اینجا کلمه مجهول calf به معنی گوساله است. اما این کلمه به معنی ماهیچه ساق پا نیز هست. همانطور که در کلمات مشابه به آن دیده می شود، امبدینگ این کلمه در context مربوط به اعضای بدن (پا) آموخته شده

است. و در محتوایی که آموزش دیده، کلمه calf و cow باهم به کار نرفته است. به همین دلیل نمیتواند بردار  
young → old را به calf → cow مپ کند.

2. الف) تابع ضرر triplet loss به صورت زیر تعریف می شود:

$$\mathcal{L}(A, P, N) = \max(d(A, P) - d(A, N) + \alpha, 0)$$

این تابع در اینجا 3 ورودی می گیرد. Y\_pred را به 3 قسمت مساوی تقسیم می کنیم تا anchor, positive, negative را بدست بیاوریم. سپس از فرمول ذکر شده استفاده می کنیم تا مقدار ضرر را محاسبه کنیم.

```
1 import tensorflow as tf
2 import tensorflow.keras.backend as K
3
4 def triplet_loss(y_true, y_pred, alpha = 0.4):
5     """
6     Implementation of the triplet loss function
7     Arguments:
8     y_true -- true labels, required when you define a loss in K
9     y_pred -- python list containing three objects:
10             anchor -- the encodings for the anchor data
11             positive -- the encodings for the positive data (similar
12             negative -- the encodings for the negative data (dissim
13     Returns:
14     loss -- real number, value of the loss
15     """
16     loss = 0
17
18     """
19     Write your code here
20     """
21     a, p, n = tf.split(y_pred, num_or_size_splits=3, axis=1)
22     dist1 = K.sqrt(K.sum(K.square(a - p), axis=-1))
23     dist2 = K.sqrt(K.sum(K.square(a - n), axis=-1))
24     return K.maximum(dist1 - dist2 + alpha, 0)
```

ب) برای ساختن مدل از چندین لایه کانولوشنی، لایه max pooling، dropout و در نهایت dense استفاده  
می کنیم تا بتوانیم برای هر تصویر ورودی ویژگی های آن را استخراج کنیم.

```
1 def embedding_pred_net(dim):
2     """
3     embedding predictions: Base network to be shared
4     """
5
6     """
7     Write your code here
8     """
9     model = Sequential()
10    model.add(Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=dim))
11    model.add(MaxPooling2D((2, 2)))
12    model.add(Dropout(0.2))
13    model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
14    model.add(MaxPooling2D((2, 2)))
15    model.add(Dropout(0.2))
16    model.add(Flatten())
17    model.add(Dense(512))
18    return model
```

پ) ابتدا باید shape داده ها را به گونه ای تغییر دهیم که بتواند وارد مدل تعریف شده شود. آن را به لیستی که 3 عضو دارد (عضو اول مجموعه تصاویر anchor، عضو دوم مجموعه تصاویر positive و عضو سوم مجموعه تصاویر negative) تبدیل می کنیم.

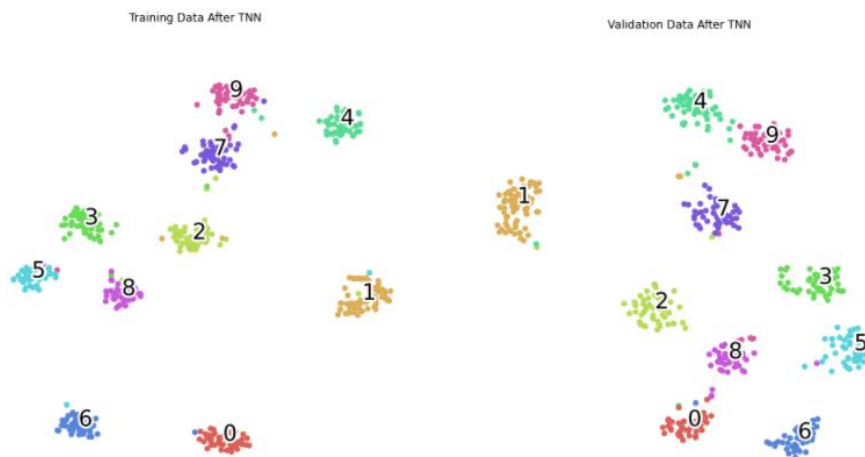
```
1 # preprocess shape of train and test set
2 shape = X_train.shape
3 label_train = np.zeros(shape[0])
4 print("labels shape", label_train.shape)
5 input_train = X_train.reshape(shape[0], shape[1], 28, 28)
6 input_train = np.swapaxes(input_train, 0, 1)
7 print("shape after swaping axis", input_train.shape)
8 input_train = list(input_train)
9 print("a p n list length", len(input_train))
10
11
12 shape = X_test.shape
13 label_test = np.zeros(shape[0])
14 input_test = X_test.reshape(shape[0], shape[1], 28, 28)
15 input_test = np.swapaxes(input_test, 0, 1)
16 print("test shape after swaping axis", input_test.shape)
17 input_test = list(input_test)
18 print("a p n list length", len(input_test))

labels shape (180000,)
shape after swaping axis (3, 180000, 28, 28)
a p n list length 3
test shape after swaping axis (3, 45000, 28, 28)
a p n list length 3
```

سپس مدل را در 15 اپیوک آموزش می دهیم. نتیجه اپیوک های پایانی به صورت زیر است:

```
Epoch 10/15
1407/1407 [=====] - 38s 27ms/step - loss: 0.0035 - val_loss: 0.1806
Epoch 11/15
1407/1407 [=====] - 37s 27ms/step - loss: 0.0039 - val_loss: 0.1642
Epoch 12/15
1407/1407 [=====] - 37s 27ms/step - loss: 0.0028 - val_loss: 0.1912
Epoch 13/15
1407/1407 [=====] - 37s 27ms/step - loss: 0.0029 - val_loss: 0.2259
Epoch 14/15
1407/1407 [=====] - 37s 27ms/step - loss: 0.0028 - val_loss: 0.2823
Epoch 15/15
1407/1407 [=====] - 37s 26ms/step - loss: 0.0028 - val_loss: 0.2154
```

برای اینکه بتوانیم بردارهای embedding را نمایش دهیم از t-SNE استفاده می کنیم تا ابعاد بردارها کاهش یابد. در تصویر زیر این بردارها دیده می شوند.



مطابق شکل، مدل به خوبی توانسته ارقام مختلف را در کلاسترهای مختلف طبقه بندی کند و نمایش دهد.

در ساختن تریپلت ها، بهترین 3 تایی ها انتخاب نشده اند. برای این که تریپلت ها به یادگیری بهتر مدل کمک کنند به صورت زیر عمل می کنیم:

انتخاب hard positive ها: برای انتخاب داده مثبت برای anchor، از میان داده هایی که با آن هم برچسب هستند، داده ای که بیشترین فاصله را با anchor دارد انتخاب می کنیم. زیرا بین آن دو داده تفاوت بیشتر است و مدل بهتر می تواند ویژگی های مربوطه را استخراج کند.

انتخاب hard negative ها: برای انتخاب داده منفی برای anchor، از میان داده هایی که برچسب شان متفاوت است، داده ای که کمترین فاصله را با anchor دارد انتخاب می کنیم. زیرا بین آن دو داده شباهت بیشتر است اما متعلق به یک کلاس نیستند.

3. ابتدا سلول های خواسته شده را پیاده سازی می کنیم. مدل را با 3 لایه کانولوشنی، لایه max pooling، لایه dropout برای جلوگیری از overfit شدن مدل، و دو لایه کاملاً متصل پیاده سازی میکنیم. خروجی لایه آخر 7 نوروں دارد چون این مسئله 7 کلاسه است.



```

1 from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, Dense, Flatten, Dropout
2 from keras.models import Model
3
4 visible = Input(shape=(28, 28, 3))
5 conv1 = Conv2D(32, kernel_size=3, activation='relu')(visible)
6 pool1 = MaxPooling2D()(conv1)
7
8 conv2 = Conv2D(64, kernel_size=3, activation='relu')(pool1)
9 pool2 = MaxPooling2D()(conv2)
10 dropout = Dropout(0.2)(pool2)
11
12 conv3 = Conv2D(128, kernel_size=3, activation='relu')(dropout)
13 pool3 = MaxPooling2D()(conv3)
14 dropout2 = Dropout(0.2)(pool3)
15
16 flat = Flatten()(dropout2)
17 hidden1 = Dense(128, activation='relu')(flat)
18 output = Dense(7, activation='softmax')(hidden1)
19 model = Model(inputs=visible, outputs=output)

```

خلاصه مدل به صورت زیر است:

Model: "model\_6"

Layer (type)	Output Shape	Param #
input_8 (InputLayer)	[None, 28, 28, 3]	0
conv2d_21 (Conv2D)	(None, 26, 26, 32)	896
max_pooling2d_20 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_22 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_21 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_9 (Dropout)	(None, 5, 5, 64)	0
conv2d_23 (Conv2D)	(None, 3, 3, 128)	73856
max_pooling2d_22 (MaxPooling2D)	(None, 1, 1, 128)	0
dropout_10 (Dropout)	(None, 1, 1, 128)	0
flatten_6 (Flatten)	(None, 128)	0
dense_12 (Dense)	(None, 128)	16512
dense_13 (Dense)	(None, 7)	903

=====  
 Total params: 110,663  
 Trainable params: 110,663  
 Non-trainable params: 0

مدل را در 50 اپیوک آموزش می دهیم (اگر بیشتر پیش برویم تنها دقت داده آموزشی بالا می رود و مدل

overfit می شود). نتایج اپیوک های پایانی به صورت زیر است:

```

219/219 [=====] - 2s 8ms/step - loss: 0.5348 - accuracy: 0.7904 - val_loss: 0.7052 - val_accuracy: 0.7438
Epoch 45/50
219/219 [=====] - 2s 8ms/step - loss: 0.5575 - accuracy: 0.7866 - val_loss: 0.6858 - val_accuracy: 0.7378
Epoch 46/50
219/219 [=====] - 2s 8ms/step - loss: 0.5373 - accuracy: 0.7981 - val_loss: 0.6762 - val_accuracy: 0.7448
Epoch 47/50
219/219 [=====] - 2s 8ms/step - loss: 0.5489 - accuracy: 0.7894 - val_loss: 0.7330 - val_accuracy: 0.7238
Epoch 48/50
219/219 [=====] - 2s 8ms/step - loss: 0.5378 - accuracy: 0.7926 - val_loss: 0.6812 - val_accuracy: 0.7468
Epoch 49/50
219/219 [=====] - 2s 8ms/step - loss: 0.5318 - accuracy: 0.7975 - val_loss: 0.6843 - val_accuracy: 0.7458
Epoch 50/50
219/219 [=====] - 2s 8ms/step - loss: 0.5324 - accuracy: 0.7963 - val_loss: 0.6785 - val_accuracy: 0.7428

```

دقت مدل را بر روی داده تست می سنجیم:

```
1 def predict_class(x):
2     pred = model.predict(x)
3     pred_class = np.argmax(pred, axis=1)
4     return np.expand_dims(pred_class, axis=1)

1 # test model accuracy on test set
2 prediction = predict_class(x_test)
3
4 test_acc = np.sum(prediction == data['test_labels']) / prediction.size
5 print("test set accuracy", test_acc)

test set accuracy 0.7496259351620947
```

الف) فرمول accuracy به صورت زیر است:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FN} + \text{FP} + \text{TN})$$

خیر، accuracy نمی تواند معیار مناسبی باشد. زیرا این دیتاست imbalanced است. تعداد داده ها با برچسب

5 در آن از سایر کلاس ها بسیار بیشتر است. یعنی مدل ویژگی های مربوط به کلاس 5 را بیشتر یاد می گیرد.

در نتیجه مدل می تواند بیشتر داده های مربوط به این کلاس را به درستی پیش بینی کند و چون این داده ها

حجم زیادی از دیتاست را تشکیل می دهد، دقت مدل بالا می رود (صورت کسر accuracy زیاد شوند).

ب) معیارها را با استفاده از کتابخانه scikit learn به صورت زیر محاسبه می کنیم.

```
1 from sklearn.metrics import classification_report
2 target_names = ['class 0', 'class 1', 'class 2', 'class 3', 'class 4', 'class 5', 'class 6']

1 train_pred = predict_class(x_train)
2 print(classification_report(data['train_labels'], train_pred, target_names=target_names))
```

	precision	recall	f1-score	support
class 0	0.65	0.65	0.65	228
class 1	0.71	0.76	0.74	359
class 2	0.70	0.67	0.69	769
class 3	0.93	0.16	0.28	80
class 4	0.74	0.64	0.69	779
class 5	0.91	0.94	0.93	4693
class 6	0.85	0.96	0.90	99
accuracy			0.85	7007
macro avg	0.78	0.68	0.69	7007
weighted avg	0.85	0.85	0.85	7007

```

1 val_pred = predict_class(x_val)
2 print(classification_report(data['val_labels'], val_pred, target_names=target_names))

```

	precision	recall	f1-score	support
class 0	0.44	0.36	0.40	33
class 1	0.40	0.60	0.48	52
class 2	0.51	0.44	0.47	110
class 3	0.00	0.00	0.00	12
class 4	0.50	0.49	0.50	111
class 5	0.86	0.88	0.87	671
class 6	0.69	0.64	0.67	14
accuracy			0.74	1003
macro avg	0.49	0.49	0.48	1003
weighted avg	0.74	0.74	0.74	1003

```

1 print(classification_report(data['test_labels'], prediction, target_names=target_names))

```

	precision	recall	f1-score	support
class 0	0.34	0.36	0.35	66
class 1	0.45	0.50	0.47	103
class 2	0.59	0.50	0.54	220
class 3	0.40	0.17	0.24	23
class 4	0.49	0.42	0.45	223
class 5	0.86	0.90	0.88	1341
class 6	0.63	0.66	0.64	29
accuracy			0.75	2005
macro avg	0.54	0.50	0.51	2005
weighted avg	0.74	0.75	0.74	2005

معیارهای محاسبه شده برای داده آزمون برای تمامی کلاس ها قابل قبول است. اما برای داده ارزیابی و آزمون، هم precision (یعنی درصد نمونه‌هایی که توسط مدل به عنوان کلاس مثبت تشخیص داده شده‌اند و درست بوده‌اند) و هم recall (یعنی درصد نمونه‌هایی که مثبت بوده‌اند و به درستی توسط مدل تشخیص داده شده‌اند) برای کلاس‌های اقلیت بسیار پایین است و مدل نتوانسته عملکرد مناسبی روی کلاس‌هایی که تعداد کمی داده داشته‌اند، کسب کند. اما این دقت نامناسب با معیار accuracy قابل سنجیدن نبود و مدل حتی بر روی داده آزمون دقت 71 درصد کسب کرده بود.

پ) برای رسم نمودار auc یک تابع تعریف میکنیم. در هر نمودار، برای هر یک از کلاس‌ها یک گراف رسم می‌کنیم.

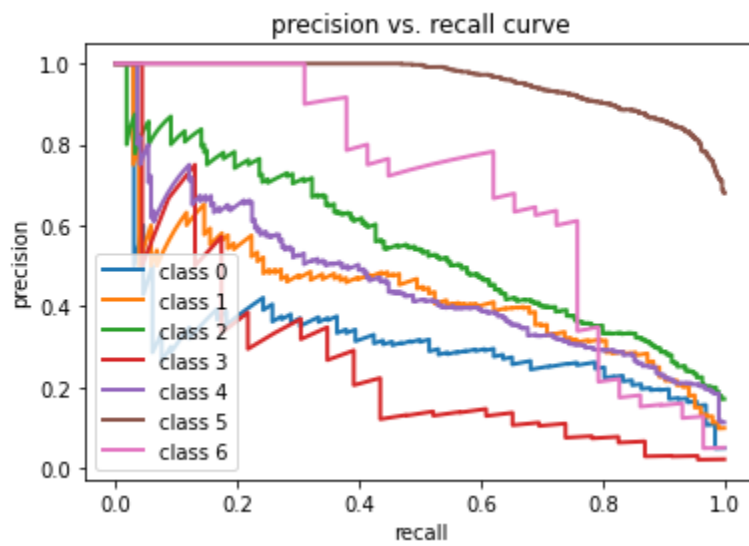
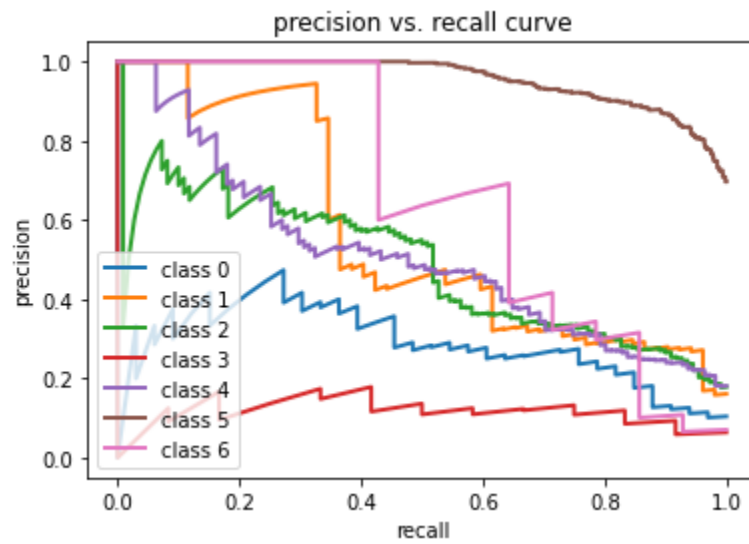
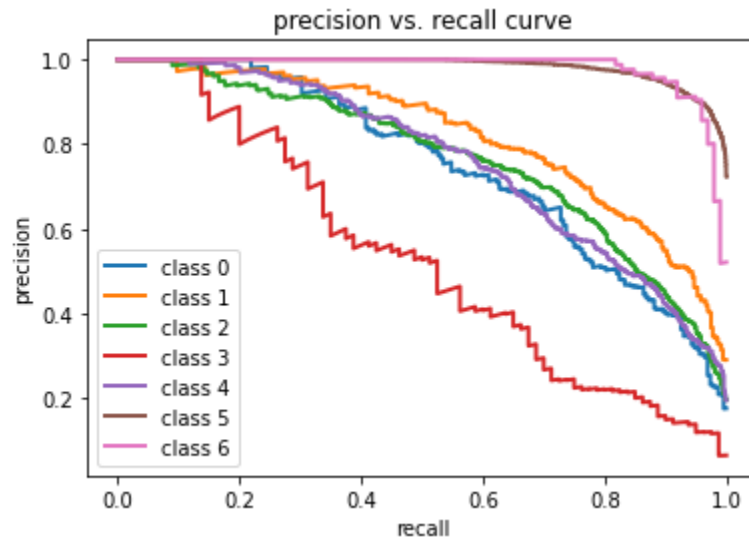
```

1 from sklearn.metrics import precision_recall_curve, roc_curve
2
3 def plot_auc(x, true_label):
4     precision = dict()
5     recall = dict()
6     for i in range(7):
7         precision[i], recall[i], _ = precision_recall_curve(true_label[:, i], model.predict(x[:, i]))
8         plt.plot(recall[i], precision[i], lw=2, label='class {}'.format(i))
9
10    plt.xlabel("recall")
11    plt.ylabel("precision")
12    plt.legend(loc="best")
13    plt.title("precision vs. recall curve")
14    plt.show()

```

در نمودار auc، برای مقادیر threshold مختلف دو معیار precision و recall را محاسبه می کنیم. این معیار می تواند یکی از معیارهای خوب ارزیابی دیتاست imbalanced باشد. هرچه سطح زیر نمودار بیشتر باشد، نشان دهنده بیشتر بودن precision, recall و یعنی کم بودن درصد false positive و false negative است.

سپس برای هر 3 نوع داده، نمودار را رسم می کنیم. نمودار داده ارزیابی و آزمون را بررسی میکنیم. سطح زیر نمودار برای کلاس 5 در نمودارها زیاد است. اما برای سایر کلاس ها نمودار auc مناسب نیست و این نشان می دهد مدل به خوبی نتوانسته ویژگی های سایر کلاس ها را یاد بگیرد.



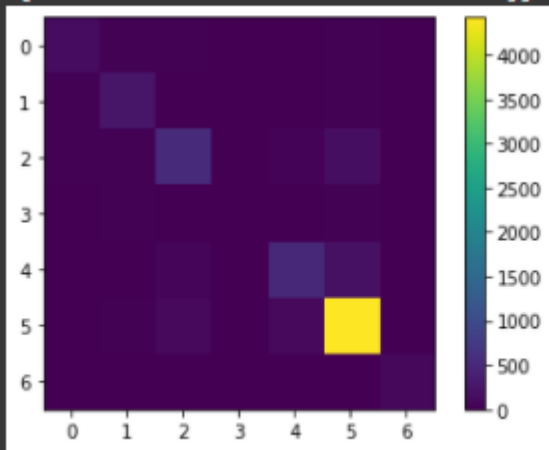
ت) برای رسم ماتریس confusion تابع را مطابق تصویر زیر تعریف می کنیم:

```
1 from sklearn.metrics import confusion_matrix
2
3 def plot_confusion(pred, true_label):
4     matrix = confusion_matrix(true_label, pred)
5     print(matrix)
6
7     plt.imshow(matrix)
8     plt.colorbar()
```

سپس آن را برای هر 3 مجموعه داده رسم می کنیم. هرچه قطر اصلی ماتریس کمرنگ تر باشد یعنی پیش بینی مدل از آن کلاس بهتر بوده. چون این ماتریس تعداد را مشخص می کند و نه تعداد، بررسی آن از precision, recall پیچیده تر است.

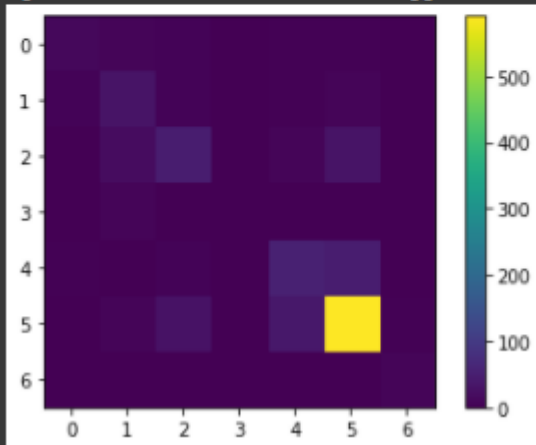
```
1 plot_confusion(train_pred, data['train_labels'])
```

```
[[ 149   31   24    0    6   18    0]
 [  28  274   16    0    6   33    2]
 [  25   21  519    1   36  162    5]
 [  13   19   14   13    3   18    0]
 [    5   11   64    0  495  202    2]
 [    8   26  108    0  119 4424    8]
 [    0    3    0    0    0    1   95]]
```



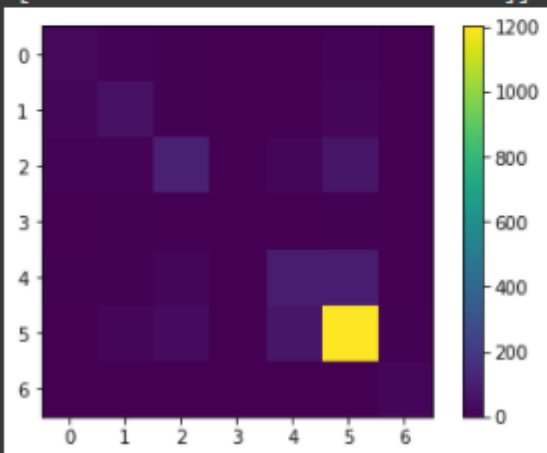
```
1 plot_confusion(val_pred, data['val_labels'])
```

```
[[ 12  9  5  1  3  3  0]
 [ 5 31  5  0  3  8  0]
 [ 2 20 48  0  8 32  0]
 [ 2  8  1  0  0  1  0]
 [ 3  1  5  0 54 48  0]
 [ 2  7 30  0 37 591  4]
 [ 1  1  0  0  2  1  9]]
```



```
1 plot_confusion(prediction, data['test_labels'])
```

```
[[ 24 14  9  3  4 12  0]
 [19 52  8  1  3 17  3]
 [13 11 109  1 19 67  0]
 [ 4  6  1  4  0  8  0]
 [ 6  7 21  1 94 92  2]
 [ 3 23 37  0 71 1201  6]
 [ 1  3  1  0  1  4 19]]
```



در ماتریس confusion هرچه اعداد روی قطر اصلی ماتریس بزرگتر باشند بهتر است. آن را می توانیم به صورت زیر نشان دهیم.

		PREDICTIVE VALUES	
		POSITIVE (1)	NEGATIVE (0)
ACTUAL VALUES	POSITIVE (1)	TP	FN
	NEGATIVE (0)	FP	TN

خانه اول یعنی داده متعلق به کلاس positive بوده و پیش بینی مدل نیز positive بوده پس مدل داده مثبت را به درستی پیش بینی کرده و tp است.

خانه دوم یعنی داده متعلق به کلاس positive بوده و پیش بینی مدل negative بوده پس مدل به اشتباه منفی پیش بینی کرده و fn است.

خانه سوم یعنی داده متعلق به کلاس negative بوده و پیش بینی مدل positive بوده پس مدل به اشتباه مثبت پیش بینی کرده و fp است.

خانه چهارم یعنی داده متعلق به کلاس negative بوده و پیش بینی مدل نیز negative بوده پس مدل به درستی منفی پیش بینی کرده و tn است.

ث) به نظر می رسد 3 معیار precision, recall, auc نمایش خوبی از عملکرد شبکه داشته اند و میزان درستی پیش بینی های مدل برای هر کلاس را به خوبی نمایش داده اند.

4. در این سوال از ابزار keras tuner استفاده می کنیم و برای مدل های پارامترهای مورد نظر را تعریف میکنیم. برای لایه کاملاً متصل از hp.Int استفاده می کنیم تا تعداد نوروں های مختلف را بسنجیم. و برای نرخ یادگیری از hp.Choice استفاده می کنیم تا مقادیر مختلف آن را امتحان کنیم.



```

1 def build_tune_model(hp):
2     model = keras.Sequential()
3     model.add(keras.layers.Flatten(input_shape=(28, 28)))
4     model.add(keras.layers.Dense(units=512, activation='relu'))
5     model.add(keras.layers.Dense(units=hp.Int('units', min_value=16, max_value=512, step=16), activation='relu'))
6     model.add(keras.layers.Dense(10))
7
8     # tune learning rate
9     alpha = hp.Choice('learning_rate', values=[0.01, 0.005, 0.001, 0.0005, 0.0001])
10    model.compile(optimizer=keras.optimizers.Adam(learning_rate=alpha),
11                  loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
12                  metrics=['accuracy'])
13
14    return model

```

سپس از hyperband استفاده می کنیم که نسبت به random search عملکرد بهتری دارد چون از early stopping استفاده می کند و در صورتی که دقت داده ارزیابی طی چند اپیوک بهتر نشود، آموزش مدل متوقف می شود.

```

1 tuner = kt.Hyperband(hypermodel=build_tune_model,
2                       objective="val_accuracy",
3                       max_epochs=30,
4                       overwrite=True,
5                       directory="./result",
6                       )
7
8 tuner.search_space_summary()

```

```

Search space summary
Default search space size: 2
units (Int)
{'default': None, 'conditions': [], 'min_value': 16, 'max_value': 512, 'step': 16, 'sampling': None}
learning_rate (Choice)
{'default': 0.01, 'conditions': [], 'values': [0.01, 0.005, 0.001, 0.0005, 0.0001], 'ordered': True}

```

سپس با استفاده از متد search مدل های مختلف با هایپر پارامترهای موجود در فضای جستجو را بررسی می کنیم و آنها را آموزش می دهیم. در ابتدای جستجو به صورت زیر است:

```

Trial 3 Complete [00h 00m 04s]
val_accuracy: 0.7946666479110718

Best val_accuracy So Far: 0.8386666774749756
Total elapsed time: 00h 00m 14s

Search: Running Trial #4

Hyperparameter |Value      |Best Value So Far
units           |416        |352
learning_rate   |0.01       |0.01
tuner/epochs    |2          |2
tuner/initial_e...|0          |0
tuner/bracket   |3          |3
tuner/round     |0          |0

```

و پس از پایان جستجو نتیجه به صورت زیر است:

```
Trial 90 Complete [00h 00m 52s]
val_accuracy: 0.8347499966621399

Best val_accuracy So Far: 0.8939999938011169
Total elapsed time: 00h 19m 55s
INFO:tensorflow:Oracle triggered exit
```

می‌توانیم نتیجه 10 مدل با بیشترین دقت ارزیابی را چاپ کنیم.

```
1 tuner.results_summary()

Results summary
Results in ./result/untitled_project
Showing 10 best trials
Objective(name='val_accuracy', direction='max')
Trial summary
Hyperparameters:
units: 96
learning_rate: 0.005
tuner/epochs: 30
tuner/initial_epoch: 10
tuner/bracket: 2
tuner/round: 2
tuner/trial_id: 16c71f938447dd919fca8e314de56910
Score: 0.8939999938011169
Trial summary
Hyperparameters:
units: 416
learning_rate: 0.01
tuner/epochs: 30
tuner/initial_epoch: 10
tuner/bracket: 3
tuner/round: 3
tuner/trial_id: f322c773e74a813af9c4d375d6133be6
Score: 0.893583357334137
Trial summary
Hyperparameters:
units: 272
learning_rate: 0.005
tuner/epochs: 30
tuner/initial_epoch: 10
```

خلاصه مدلی که بهترین نتیجه را کسب کرده به صورت زیر است:

```
1 models = tuner.get_best_models(num_models=2)
2 best_model = models[0]
3 best_model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 512)	401920
dense_1 (Dense)	(None, 96)	49248
dense_2 (Dense)	(None, 10)	970

```
=====
Total params: 452,138
Trainable params: 452,138
Non-trainable params: 0
```

مدلی که دارای 96 نورون در لایه کاملاً متصل با نرخ یادگیری 0.005 بوده بهترین دقت را کسب کرده است. دقت آن 89 درصد است که به نسبت مدل پایه (512 نورون در لایه کاملاً متصل با نرخ یادگیری 0.01)، 4 درصد پیشرفت داشته. با کمتر کردن تعداد نورون ها، ظرفیت یادگیری شبکه کمتر می شود و مدل کمتر به حفظ الگوها می پردازد (هرچند ممکن است دقت داده آموزشی کمتر شود، اما دقت آزمون بهتر خواهد شد). همچنین با کاهش نرخ یادگیری عملکرد بهبود یافته است. با کم کردن نرخ یادگیری، سرعت آموزش کمتر خواهد شد و هزینه محاسباتی بیشتر می شود اما احتمال اینکه مدل در نقطه sub optimal گیر کند کمتر می شود و مدل بهتر آموزش می بیند.

## منابع

[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification\\_report.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html)

<https://stackoverflow.com/questions/56090541/how-to-plot-precision-and-recall-of-multiclass-classifier>

[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\\_matrix.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html)

<https://medium.com/datasciencestory/performance-metrics-for-evaluating-a-model-on-an-imbalanced-data-set-1feeab6c36fe>