

به نام خدا



تمرین سری چهاردهم درس یادگیری عمیق

دکتر محمدی

محمد یارمقدم

۹۶۴۶۲۱۰۴

سوال اول)

الف) برای این سوال ۱۰ کلمه زیر را به عنوان ورودی به شبکه دادم:

Pen, club, expensive, vote, lion, city, happy, apple, fruit, pretty

پس از ران شدن، خروجی زیر بدست آمد:

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10
pen	(ballpoint, 0.61)	(pens, 0.56)	(pencil, 0.5)	(ink, 0.49)	(le, 0.48)	(duick, 0.48)	(quill, 0.47)	(jean-marie, 0.45)	(argyl, 0.44)	(sovann, 0.44)
club	(clubs, 0.79)	(football, 0.69)	(league, 0.67)	(soccer, 0.63)	(rugby, 0.6)	(team, 0.6)	(playing, 0.6)	(side, 0.6)	(professional, 0.59)	(players, 0.58)
expensive	(costly, 0.78)	(pricey, 0.72)	(cheaper, 0.71)	(cost, 0.68)	(inexpensive, 0.67)	(cheap, 0.67)	(less, 0.66)	(afford, 0.65)	(costs, 0.62)	(priced, 0.62)
vote	(votes, 0.85)	(voting, 0.85)	(election, 0.8)	(elections, 0.78)	(voters, 0.77)	(voted, 0.77)	(polls, 0.74)	(ballot, 0.73)	(majority, 0.7)	(referendum, 0.7)
lion	(elephant, 0.6)	(wolf, 0.54)	(bear, 0.54)	(dragon, 0.53)	(leopard, 0.51)	(beast, 0.5)	(tamarin, 0.5)	(dog, 0.49)	(golden, 0.49)	(monkey, 0.48)
city	(cities, 0.78)	(town, 0.73)	(downtown, 0.71)	(where, 0.65)	(area, 0.65)	(mayor, 0.64)	(in, 0.63)	(residents, 0.63)	(towns, 0.61)	(capital, 0.6)
happy	(glad, 0.74)	('m, 0.74)	(always, 0.73)	(really, 0.72)	(everyone, 0.72)	(pleased, 0.71)	(wish, 0.7)	(everybody, 0.7)	(good, 0.69)	(feel, 0.69)
apple	(iphone, 0.63)	(microsoft, 0.61)	(intel, 0.6)	(macintosh, 0.6)	(ipod, 0.59)	(ibm, 0.59)	(ipad, 0.59)	(software, 0.57)	(google, 0.56)	(itunes, 0.55)
fruit	(fruits, 0.84)	(vegetables, 0.68)	(citrus, 0.67)	(berries, 0.65)	(vegetable, 0.64)	(juice, 0.62)	(mango, 0.61)	(grape, 0.61)	(grapes, 0.6)	(bananas, 0.6)
pretty	(quite, 0.84)	(really, 0.82)	(very, 0.77)	(thing, 0.76)	('re, 0.75)	(good, 0.74)	('m, 0.74)	(looks, 0.74)	(obviously, 0.73)	(things, 0.73)

در این مدل میزان شباهت هر کلمه با ۱۰ کلمه موجود در جدول مشخص شده است. به طور مثال در دو ردیف pen و club کلمات و درصدهای مناسبی نمایش داده شده است. اما در سطر lion به طور مثال میزان شباهت elephant از wolf بیشتر نمایش داده است که به نظر اشتباه می‌رسد زیرا در ابعاد و اندازه و وحشی‌گری شباهت wolf باید بیشتر باشد و ارتباط معنایی آن دو بیشتر است. همچنین به طور مثال tamarin در کلمات مشابه lion نمایش داده شده است که هیچ ارتباط معنایی درستی ندارد. City هم کلمات مناسبی دارد ولی می‌توانست لغات بهتری هم یافت شود. به طور مثال in و where بیشتر لغاتی است که در جمله معنا دارد و به تنهایی سخت ارتباط معنایی آن را می‌توان مشخص کرد. اما بدترین لغات یافت شده برای سطر pretty است که کلمات عجیب و بی ارتباطی نمایش داده شده است.

ب) در این مورد سه تایی‌های زیر به عنوان ورودی داده شد. کلمه اول ورودی، کلمه دوم کلمه نزدیک قابل انتظار و کلمه سوم کلمه دور قابل انتظار است.

pretty, beautiful, ugly

good, nice , bad

active, energetic, tired

go, leave, come

fast, rapid, slow

کدر خروجی جدول زیر حاصل شد:

	Near Word	Far Word
pretty	(beautiful, 0.38)	(ugly, 0.44)
good	(nice, 0.34)	(bad, 0.29)
active	(energetic, 0.59)	(tired, 0.84)
go	(leave, 0.24)	(come, 0.14)
fast	(rapid, 0.46)	(slow, 0.25)

به جز سطر **active** بقیه موارد نتایج با اعداد پایینی کسب شد. به طور مثال برای فعلی مثل **go** کلمات نزدیک آن احتمالا حروف اضافه بعد از آن است یا قید و صفاتی که از آن ساخته میشود چون فعل هم معنی آن کم شباهت دارد. هم چنین در سطر **active** کلمه **tired** که متضاد آن است از هم معنی آن بیشتر شباهت داشت که مورد عجیب و اشتباهی به نظر می‌رسد.

ج) فاصله اول معنا های بهتری را تشکیل داده است. زیرا به طور مثال در سطر های **actor** یا **doctor** کلمات **comedian** و **physician** به عنوان بهترین کلمه نمایش داده شده است که نام حرفه و شغل کلمه اول است و رابطه با سه کلمه دیگر دارد. دلیل آن است که در نوع اول کلمه اول و دوم با هم برای درک رابطه معنایی در نظر گرفته شده است اما در نوع دوم کلمه اول و سوم که باعث بروز این مورد میشود.

د) در این نمودار سه دسته تشکیل شده است. **King** و **queen** با هم، **actor** و **actress** با هم و **woman** و **men** و **girl** و **boy** نیز با هم هستند. در ابتدا مشخص میشود که دسته‌بندی ها بر اساس حوزه معنایی و شغلی تشکیل شده است و جنسیت های متفاوت نیز در کنار هم قرار گرفتند.

حال الگو را میتوان به طور زیر تفسیر کرد. بردار بین **king** و **queen** در همان راستای **actor** و **actress** است. پس میتوان فهمید جهت این بردار جنسیت را مشخص می‌کند. تایید آن را در جهت بردار **man** و **woman** میتوان دید. بردار **girl** و **woman** هم در جهت **boy** و **man** است. پس این جهت نیز فاصله سنی را مشخص می‌کند.

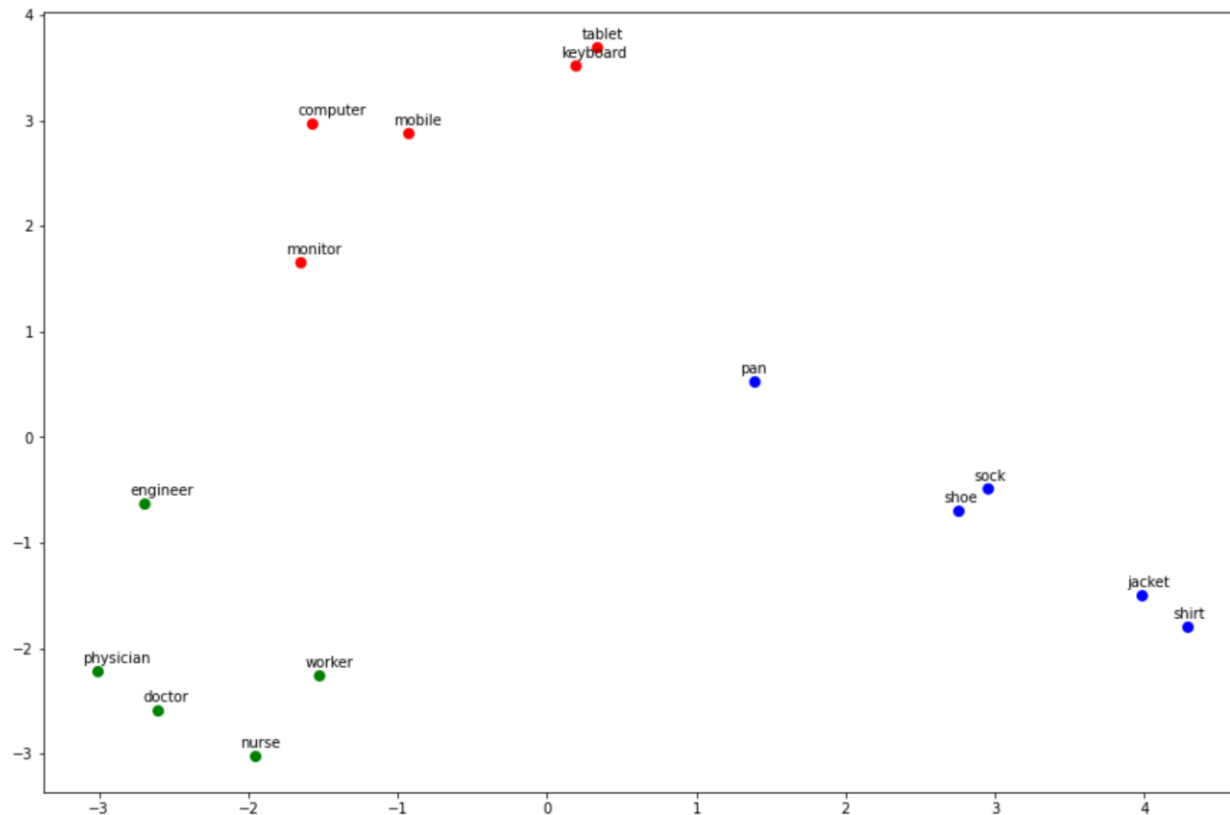
ه) در این مورد کشورها و پایتختشان داده شد که با هم در کنار یکدیگر قرار گرفتند و دو نوع رابطه بین آنها ایجاد شد. **Ankara** و **Turkey** مثل **Iraq** و **Baghdad** شد و **Russia** و **Moscow** نیز مثل آنها شد و بقیه موارد نیز جهت یکسانی گرفتند. همچنین **Canada** و **England** نیز مثل هم شدند.

و) در این مورد سه دسته‌بندی کلی ایجاد شده است. اما جهت برخی بردار های الگو هایی دارد. بردار تضاد بین **red** و **green** مشابه **cat** و **mouse** و یا **cat** و **dog** که تضاد موجود در بین آنها را بیان می‌کند. حال این بردار ها در بین کشور های هم مشابه شده است. جهت بین **lion** و **cow** به عنوان طعمه بین کشور های نیز تکرار شده است که نکته جالبی بود.

ک) در این بخش سه دسته‌بندی شغل، پوشاک و تکنولوژی و لوازم جانبی داده شد.

```
words_group1 = ['doctor', 'nurse', 'physician', 'worker', 'engineer']
words_group2 = ['shoe', 'sock', 'shirt', 'jacket', 'pan']
words_group3 = ['computer', 'mobile', 'tablet', 'monitor', 'keyboard']
```

نتیجه به صورت زیر شد که تقریبا سه دسته‌بندی خوب شکل گرفت:



الگو تا حدود خوبی مشابه است و روابطی مشابه میتوان یافت.

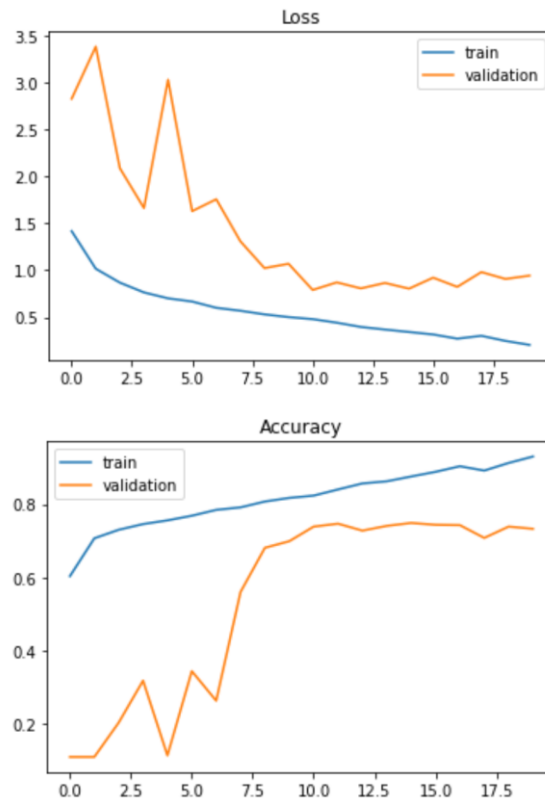
ل) کلمه موردنظر همان COW باید باشد چون کلمه متفاوتی برای گاو جوان و پیر نداریم.

سوال سوم

ابتدا مدل را روی یک شبکه که در تمرین قبل مورد استفاده قرار گرفت و نتایج خوبی را ارائه داد آموزش دادم و نتایج زیر کسب شد:

```
Epoch 16/20
55/55 [=====] - 2s 39ms/step - loss: 0.3161 - accuracy: 0.8884 - val_loss: 0.9205 - val_accuracy: 0.7438
Epoch 17/20
55/55 [=====] - 2s 40ms/step - loss: 0.2716 - accuracy: 0.9037 - val_loss: 0.8226 - val_accuracy: 0.7428
Epoch 18/20
55/55 [=====] - 2s 40ms/step - loss: 0.3020 - accuracy: 0.8918 - val_loss: 0.9799 - val_accuracy: 0.7079
Epoch 19/20
55/55 [=====] - 2s 40ms/step - loss: 0.2476 - accuracy: 0.9127 - val_loss: 0.9068 - val_accuracy: 0.7388
Epoch 20/20
55/55 [=====] - 2s 40ms/step - loss: 0.2052 - accuracy: 0.9302 - val_loss: 0.9422 - val_accuracy: 0.7328
```

همچنین نمودار خطا و دقت رسم شد تا روند یادگیری مدل مورد ارزیابی قرار گیرد.



الف) معیار دقت برای ارزیابی مدل مناسب نیست. زیرا داده تست در همه لیبل ها به اندازه مناسب و یکسان وجود ندارد و ممکن مدل یک لیبل را به خوبی بشناسد و تست هم روی این لیبل از داده های تست انتخاب شود و دقت خوبی هم نشان بدهد اما در سایر لیبل ها یادگیری خوب صورت نگرفته باشد.

ب) با استفاده از متد های موجود در کتابخانه sklearn میتوان به راحتی معیار های recall و precision را محاسبه کرد. در متد calculate_variables این معیار ها محاسبه شد. هم ماتریس confusion نیز رسم شد.

داده های آموزش:

class 0	0.00	0.00	0.00	228
class 1	0.00	0.00	0.00	359
class 2	0.00	0.00	0.00	769
class 3	0.00	0.00	0.00	80
class 4	0.00	0.00	0.00	779
class 5	0.67	1.00	0.80	4693
class 6	0.00	0.00	0.00	99
accuracy			0.67	7007
macro avg	0.10	0.14	0.11	7007
weighted avg	0.45	0.67	0.54	7007

داده های تست:

	precision	recall	f1-score	support
class 0	0.00	0.00	0.00	66
class 1	0.00	0.00	0.00	103
class 2	0.00	0.00	0.00	220
class 3	0.00	0.00	0.00	23
class 4	0.00	0.00	0.00	223
class 5	0.67	1.00	0.80	1341
class 6	0.00	0.00	0.00	29
accuracy			0.67	2005
macro avg	0.10	0.14	0.11	2005
weighted avg	0.45	0.67	0.54	2005

داده های اعتبارسنجی:

	precision	recall	f1-score	support
class 0	0.00	0.00	0.00	33
class 1	0.00	0.00	0.00	52
class 2	0.00	0.00	0.00	110
class 3	0.00	0.00	0.00	12
class 4	0.00	0.00	0.00	111
class 5	0.67	1.00	0.80	671
class 6	0.00	0.00	0.00	14
accuracy			0.67	1003
macro avg	0.10	0.14	0.11	1003
weighted avg	0.45	0.67	0.54	1003

Precision: نسبت True Positive به مجموع True Positive و False Positive است. در واقع نسبت True Positive به تمامی پیشبینیهای Positive مفهوم آن این است که مدل در پیشبینیهایی که گفته Positive است چقدر دقیق بوده است.

Recall: نسبت True Positive به مجموع حالتهایی که واقعا True هستند میگویند. در واقع بیانگر این است که مدل ما چه تعداد از دادههایی را که واقعا Positive هستند را درست Positive پیشبینی کرده است.

در هر سه مورد در کلاس پنجم که داده های آن از سایر کلاس ها بیشتر هم مقدار precision و هم مقدار recall بالاست که نشان دهنده نتایج خوبی نیست. Recall برابر ۱ است. این نشان از آن دارد که مدل به طور اشتباه منفی پیشبینی اشتباه نداشته. با توجه به تعداد کم سایر کلاس ها قابل پیش بینی بود. اما در precision مقدار ۰.۶۷ دارد که نشان از آن دارد که به طور اشتباه تعدادی را متعلق به این کلاس معرفی کرده است که دلیل تعداد کم کلاس های دیگر است که به این دلیل نمیتواند گونه های مختلف سایر کلاس ها را بیاموزد.

کلاس های دیگر recall و precision برابر صفر دارند. Recall برابر صفر یعنی هیچ پیش بینی درستی نداشتیم که نشان از یادگیری خیلی پایین شبکه برای کلاس هایی که داده ی زیادی ندارند است. Precision نیز صفر است که یعنی صورت و مخرج برابر صفر است که چون هیچ چیزی را در سن پیش بینی نکرده صفر در نظر میگیریم.

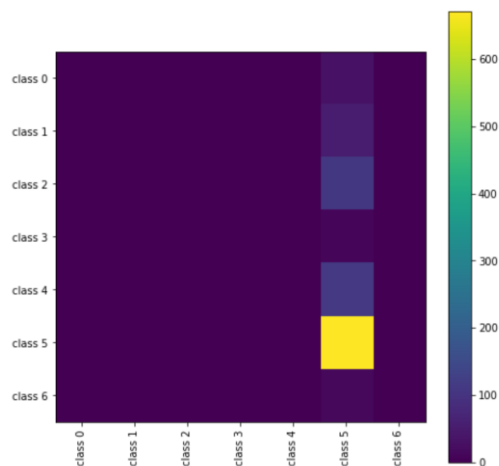
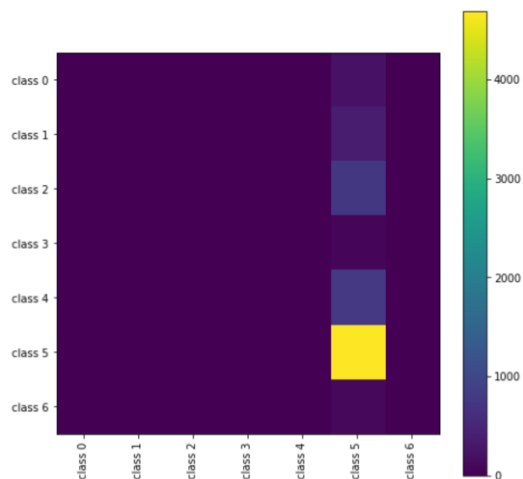
ROC یک منحنی احتمال است و **AUC** نشان دهنده درجه یا معیار تفکیک پذیری است. این نشان می دهد که مدل چقدر می تواند بین کلاس ها تمایز قائل شود. هرچه **AUC** بالاتر باشد، مدل در پیش بینی کلاس های ۰ به عنوان ۱ و کلاس های ۱ به عنوان ۰ بهتر است. بر اساس قیاس، هرچه **AUC** بالاتر باشد، مدل در تشخیص بیماران مبتلا به بیماری و بدون بیماری بهتر است. نمودار ها در نوت بوک رسم شده اند.

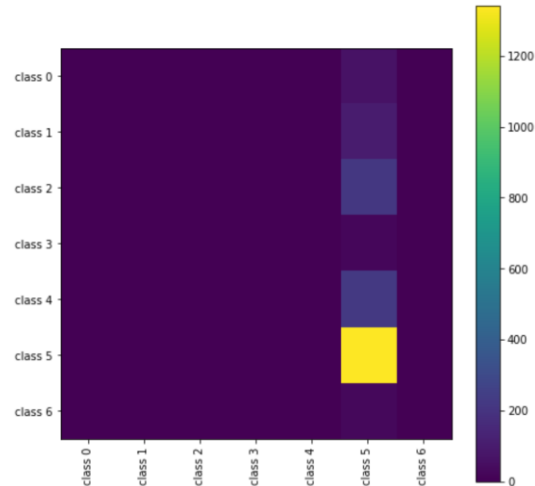
ت **confusion matrix** در واقع یک اندازه گیری عملکرد برای مسئله طبقه بندی یادگیری ماشین است که در آن خروجی می تواند دو یا چند کلاس باشد. این یک جدول با ۴ ترکیب مختلف از مقادیر پیش بینی شده و واقعی است.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

در این جدول چهار مقدار true positive، false positive، false negative و true negative وجود دارد که با توجه به عملکرد مدل مقدار دهی میشود.

ماتریس های رسم شده به شکل زیر می باشند:





با ترتیب ماتریس ها مربوط به داده های آموزش، ولیدیشن و تست است.

هر چه طیف رنگی پر رنگ تر باشد به این مفهوم است که خطای موجود مابین دو کلاس بیشتر بوده است. طبق این مورد نیز عملکرد مدل مناسب نبوده. چون دو کلاس رنگ پر رنگی دارند و کاملاً با هم اشتباه گرفته شده اند.

ث) recall برای این مسئله مناسب است زیرا اگر کسی که یک بیماری را بیماری دیگر برای آن معرفی کنیم اتفاق مناسبی نیست. معیار F1 که Precision و recall هست بهترین معیار است زیرا هر مورد را در دل خود جای میدهد.
(منابع)

<https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>

https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

<https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

سوال چهارم)

در این سوال ابتدا شبکه داده شده را ران کردم. نتایج زیر بدست آمد:

```
1500/1500 [=====] - 10s 7ms/step - loss: 0.3043 - accuracy: 0.8909 - val_loss: 0.6379 - val_accuracy: 0.8537
Epoch 47/50
1500/1500 [=====] - 10s 6ms/step - loss: 0.3096 - accuracy: 0.8876 - val_loss: 0.5883 - val_accuracy: 0.8555
Epoch 48/50
1500/1500 [=====] - 10s 7ms/step - loss: 0.3021 - accuracy: 0.8892 - val_loss: 0.5692 - val_accuracy: 0.8537
Epoch 49/50
1500/1500 [=====] - 10s 7ms/step - loss: 0.3053 - accuracy: 0.8873 - val_loss: 0.6248 - val_accuracy: 0.8473
Epoch 50/50
1500/1500 [=====] - 11s 7ms/step - loss: 0.3028 - accuracy: 0.8898 - val_loss: 0.5883 - val_accuracy: 0.8604
```

سپس هدف بهینه کردن مقدار هایپرپارامتر های شبکه است. برای این کار مطابق تمرین قبل که داشتیم یک tuner_model ساخته و لایه های گفته شده را با رنج مورد نظر در آن به شکل زیر می سازیم:


```
def build_tuner_model(hp):
    model = keras.models.Sequential()
    model.add(keras.layers.Flatten(input_shape=(28, 28)))
    model.add(keras.layers.Dense(hp.Int('dense_units', min_value=16, max_value=512, step=16)))
    model.add(tf.keras.layers.Activation('relu'))
    model.add(keras.layers.Dense(10))
    learning_rate = hp.Choice('learning_rate', values=[0.01, 0.005, 0.001, 0.0005, 0.0001])
    model.compile(optimizer=keras.optimizers.Adam(learning_rate=learning_rate),
                  loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
                  metrics=['accuracy'])

    return model
```

سپس مدل را ران میکنیم تا مقدار بهینه برای پارامترهای شبکه تعیین شود. در 5 تریال این امر در epoch های ۲ تایی انجام می شود تا بهترین نتیجه از ران های مختلف مشخص شود. مدل استفاده شده در tuner نیز یک هایپرپارامتر به شکل زیر است:

```
tuner = Hyperband(
    build_tuner_model,
    objective='val_accuracy',
    max_epochs=5,
    executions_per_trial=5,
)
```

نتیجه بهترین حالت tune به صورت زیر است:

```
Trial summary
Hyperparameters:
dense_units: 320
learning_rate: 0.001
tuner/epochs: 5
tuner/initial_epoch: 0
tuner/bracket: 0
tuner/round: 0
Score: 0.8840999960899353
```

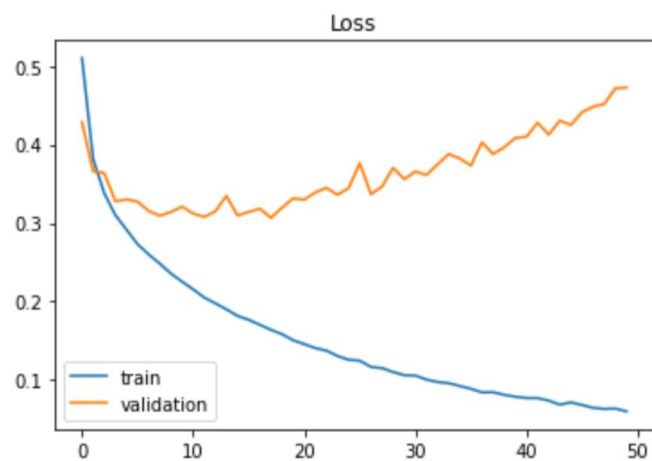
سپس مقادیر مشخص شده در نوت بوک را دوباره به مدل داده و آن را دوباره ران کردم. نتایج زیر بدست آمد:

```
Epoch 47/50
1500/1500 [=====] - 5s 3ms/step - loss: 0.0644 - accuracy: 0.9758 - val_loss: 0.4483 - val_accuracy: 0.8979
Epoch 48/50
1500/1500 [=====] - 6s 4ms/step - loss: 0.0631 - accuracy: 0.9766 - val_loss: 0.4517 - val_accuracy: 0.8984
Epoch 49/50
1500/1500 [=====] - 5s 3ms/step - loss: 0.0635 - accuracy: 0.9770 - val_loss: 0.4716 - val_accuracy: 0.8916
Epoch 50/50
1500/1500 [=====] - 6s 4ms/step - loss: 0.0600 - accuracy: 0.9783 - val_loss: 0.4726 - val_accuracy: 0.8939
```

همانطور که مشاهده می شود دقت روی داده تست حدود ۵ درصد نسبت به مدل پایه افزایش یافته است. دلیل این بهبود این است که مدل در حالت پایه دارای ۳۲۰ نورون در لایه میانی و نرخ آموزش ۰.۰۰۱ بود که این دو مقدار ظرفیت شبکه را بالا برده بودند و باعث می شدند شبکه

در همان Epoch های ابتدایی Converge شود و از اواسط آموزش دیگر بهبودی نداشته باشیم Underfit با کاهش نرخ آموزش به ۰,۰۰۱ مشکل Underfit را حل کردیم و همچنین باعث شدیم overfit مدل در epoch های بالایی انجام شود که بهبود محسوب میشود.

نمودار دقت و خطا نیز به صورت زیر است:



<Figure size 432x288 with 0 Axes>

