



به نام خدا
دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر



درس یادگیری ماشین تمرین چهارم

نام و نام خانوادگی	امیرحسین پورداود
شماره دانشجویی	۸۱۰۱۰۱۱۲۰
تاریخ ارسال گزارش	۱۴۰۲.۱۰.۱۷

فهرست

پاسخ ۱ - پرسش	۲
۱-۱. قسمت ۱ (معماری شبکه عصبی)	۲
۲-۱. قسمت ۲ (آموزش شبکه عصبی)	۳
پاسخ ۲ - پرسش	۵
۱-۲. قسمت ۱	۵
۲-۲. قسمت ۲	۶
پاسخ ۳ - پرسش	۷
۱-۳. قسمت ۱	۷
۲-۳. قسمت ۲	۸
پاسخ ۴ - پرسش	۹
پاسخ ۵ - شبیه سازی	۱۰
۱-۵. قسمت ۱ (MLP)	۱۰
۲-۵. قسمت ۲ (CNN)	۱۱
۳-۵. قسمت ۳ (Mobile Net)	۱۲
پاسخ ۶- شبیه سازی	۱۴

پاسخ ۱ - پرسش

۱-۱. قسمت ۱ (معماری شبکه عصبی)

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

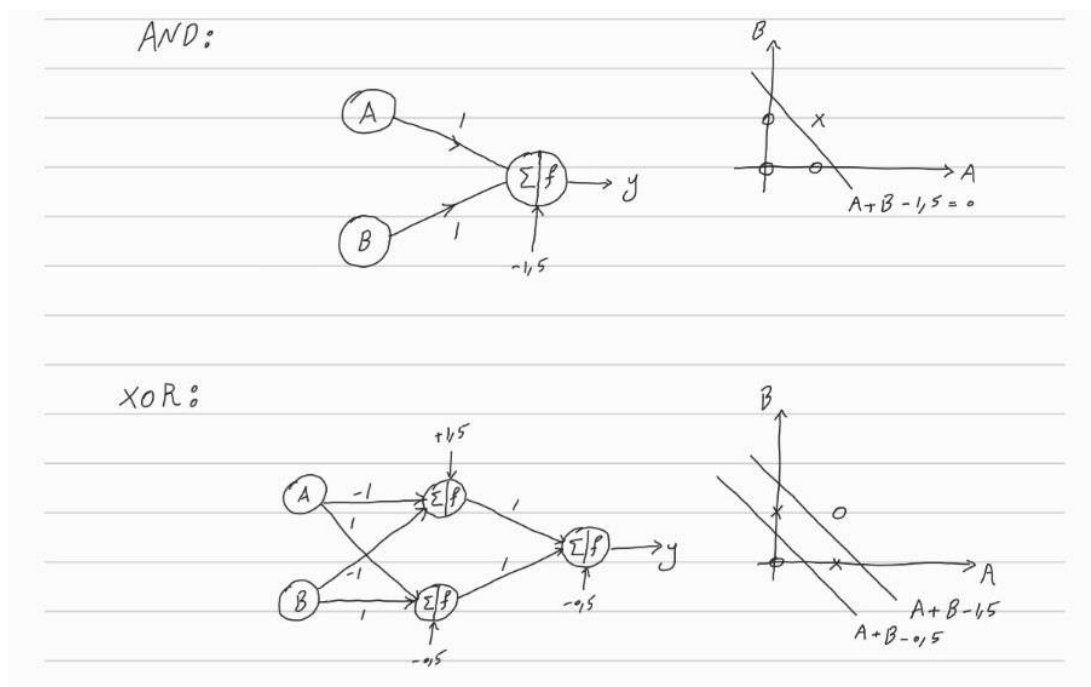
۱-۱-۱. چند لایه استفاده میکنید؟ چرا؟

۱-۱-۲. چند گره در هر لایه استفاده میکنید؟ چرا؟

همانطور که در شکل قسمت سوم قابل مشاهده است، در قسمت AND ما میتوانیم تنها با یک خط کلاس ها را از یکدیگر جدا کنیم، بنابراین تنها نیاز به یک گره برای ایجاد خط و یک لایه برای جدا سازی داریم.

اما برای XOR، ما با دو خط باید جدا سازی را انجام دهیم، بنابراین به ازای هر خط به یک گره نیاز داریم که در مجموع دو گره میشود، پس در لایه اول دو گره داریم، در نهایت باید این دو گره را با هم Or کنیم تا جداسازی کامل شود، بنابراین در مجموع به دو لایه نیاز خواهیم داشت.

۱-۱-۳. شکل شبکه عصبی تان را رسم کنید.



۲-۱. قسمت ۲ (آموزش شبکه عصبی)

۲-۱-۱. الگوریتم به عقب انتشار خطا به دنبال چه چیزی می‌گردد؟

الگوریتم پس انتشار یک تکنیک یادگیری تحت نظارت است که در آموزش شبکه های عصبی مصنوعی استفاده می شود. هدف اصلی آن به حداقل رساندن خطا بین خروجی پیش بینی شده و خروجی هدف واقعی با تنظیم وزن اتصالات شبکه است. به عبارت دیگر، هدف آن بهینه سازی پارامترهای شبکه (وزن ها و سوگیری ها) برای بهبود عملکرد آن در یک کار معین است.

۲-۲-۱. چگونه وزنه های اتصالات شبکه ها را مقداردهی اولیه می کنید؟

مقدار دهی اولیه وزن های شبکه عصبی یک گام مهم است که می تواند بر روند آموزش تأثیر بگذارد. مقداردهی اولیه وزن مناسب به جلوگیری از مشکلاتی مانند vanishing یا exploding گرادیان ها کمک می کند، که می تواند مانع همگرایی مدل در طول تمرین شود. روش های متداول برای مقداردهی اولیه وزن عبارتند از:

- مقدار دهی اولیه: تنظیم همه وزن ها روی صفر، اما معمولاً این کار توصیه نمی شود زیرا می تواند منجر به مشکلات تقارن شود.

- مقدار دهی اولیه تصادفی: اختصاص مقادیر تصادفی کوچک به وزن ها. این یک رویکرد رایج و مؤثر است و تغییرات شامل مقداردهی اولیه یا گاوسی اولیه و مقداردهی اولیه یکنواخت است.

- مقداردهی اولیه Xavier/Glorot وزن ها را بر اساس تعداد واحدهای ورودی و خروجی یک لایه، با هدف حفظ تعادل بین گرادیان های در حال انفجار و ناپدید شدن مقیاس می کند.

- مقدار دهی اولیه یلهشبه به Xavier، اما با یک ضریب مقیاس بندی کمی متفاوت، که اغلب با توابع فعال سازی مانند ReLU استفاده می شود.

۲-۲-۱. مراحل انجام شده توسط الگوریتم به عقب انتشار خطا در یک دوره epoch را توضیح دهید

یک دوره به یک گذر کامل از کل مجموعه داده آموزشی اشاره دارد. الگوریتم پس انتشار شامل چندین مرحله در طول هر دوره است:

۱. Forward Pass: داده های ورودی به شبکه تغذیه می شود و محاسبات لایه به لایه از طریق لایه های پنهان انجام می شود تا خروجی پیش بینی شده تولید شود.
۲. خروجی پیش بینی شده با خروجی هدف واقعی با استفاده از تابع ضرر مقایسه می شود که خطای بین مقادیر پیش بینی شده و واقعی را اندازه گیری می کند.

۳. Backward Pass (Backpropagation): گرادیان از دست دادن با توجه به وزن و بایاس با استفاده از قانون زنجیره ای حساب محاسبه می شود. این شامل محاسبه این است که اگر هر وزن و بایاس تعدیل شود چقدر از دست دادن تغییر می کند.

۴. گرادیان های محاسبه شده برای به روز رسانی وزن ها و بایاس ها در جهت مخالف گرادیان استفاده می شود تا تلفات به حداقل برسد. این معمولاً با استفاده از یک الگوریتم بهینه سازی مانند نزول گرادیان تصادفی (SGD) یا یکی از انواع آن انجام می شود.

۵. مراحل ۱-۴ برای تعداد دوره های از پیش تعریف شده یا تا زمان همگرایی تکرار می شوند، وزن ها و سوگیری ها به طور مکرر تنظیم می شوند تا از دست دادن کلی در مجموعه داده آموزشی به حداقل برسد.

۶. پس از آموزش، عملکرد مدل اغلب بر روی یک مجموعه اعتبارسنجی جداگانه ارزیابی می شود تا اطمینان حاصل شود که به خوبی به داده های دیده نشده تعمیم می یابد. علاوه بر این، مدل را می توان بر روی یک مجموعه تست برای ارزیابی عملکرد کلی آن آزمایش کرد.

۲-۱. قسمت ۱

برای یافتن توپ سنگین‌تر از میان ۹ توپ فلزی با کمترین تعداد آزمایش ممکن، می‌توان از یک استراتژی جستجو مانند باینری استفاده کرد. در اینجا یک استراتژی است که فقط به ۳ با وزن کردن نیاز دارد:

۱. مرحله اول (۳ در مقابل ۳):

- ۹ توپ را به سه گروه سه تایی تقسیم می‌کنیم.
- دو تا از گروه‌ها را با هم وزن می‌کنیم.
- اگر یک طرف آن سنگین‌تر است، با آن سه توپ به مرحله بعدی می‌رویم. اگر تعادل داشتند، با سه توپ باقی مانده به مرحله بعدی می‌رویم.

۲. مرحله دوم (۱ در مقابل ۱):

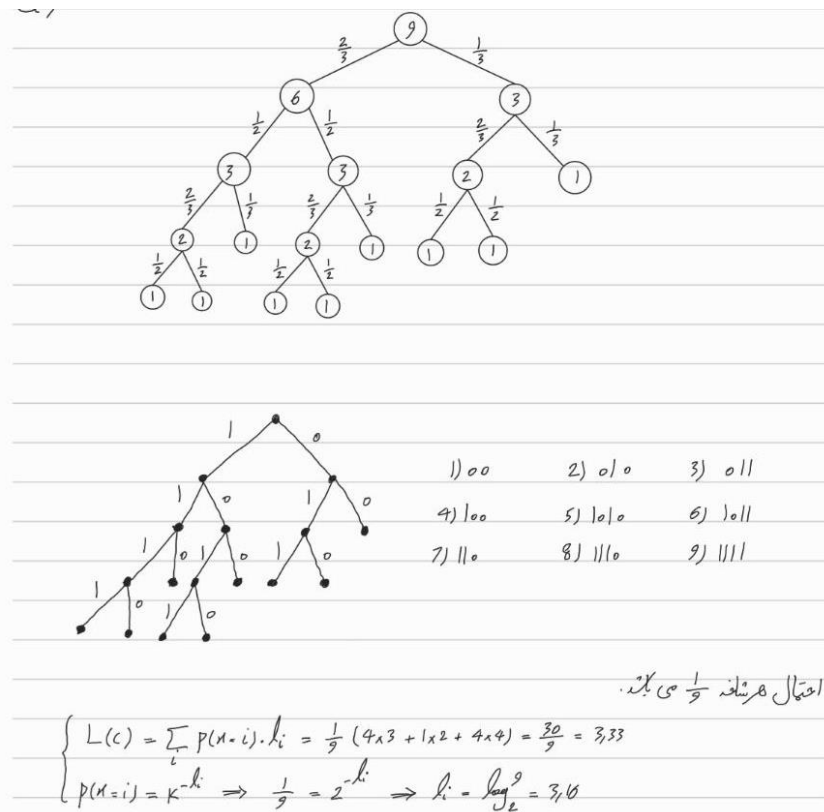
- سه توپ مشخص شده در وزن کشی اول را برمی‌داریم.
- یک توپ را در برابر توپ دیگر از این گروه وزن می‌کنیم.
- اگر یک طرف سنگین‌تر باشد، توپ سنگین‌تر را پیدا کرده‌ایم. اگر تعادل داشته باشند، توپ باقی مانده سنگین‌تر است.

۳. وزن سوم (۱ در مقابل ۱):

- اگر در وزن کشی دوم توپ سنگین‌تری را شناسایی کردیم، از توپ باقی مانده از آن گروه استفاده می‌کنیم.
- توپ باقیمانده را با هر یک از شش توپ دیگر که در وزن کردن‌های قبلی استفاده نشده است وزن می‌کنیم.
- اگر یک طرف سنگین‌تر باشد، توپ سنگین‌تر را پیدا کرده‌ایم. اگر تعادل داشته باشند، توپ باقی مانده سنگین‌تر است.
-

این استراتژی تضمین می‌کند که توپ سنگین‌تر را تنها با سه بار وزن کردن می‌توانیم شناسایی کنیم. حال بهینه بودن اطلاعات آن را تحلیل می‌کنیم:

تعداد کل نتایج ممکن (پیکربندی توپ‌های سنگین‌تر و سبک‌تر) $2^9 = 512$ است.



این استراتژی یک نتیجه منحصر به فرد را برای هر پیکربندی با حداکثر سه وزن، که مربوط به یک کد باینری به طول سه است، ارائه می دهد. میانگین طول کد $H(X) \leq 3$ است که در آن $H(X)$ آنتروپی است. بنابراین، استراتژی از نظر اطلاعاتی بهینه است، زیرا میانگین طول کد مورد نیاز برای یافتن توپ سنگین تر به حداقل می رسد و این حداقل با استراتژی پیشنهادی به دست می آید.

۲-۲. قسمت ۲

الگوریتم ID3 که معمولاً برای ساخت درخت های تصمیم گیری در یادگیری ماشین استفاده می شود، ممکن است لزوماً درخت تصمیم گیری بهینه را برای مسئله ۹ توپ فلزی ایجاد نکند. الگوریتم ID3 درخت های تصمیم را بر اساس کسب اطلاعات در هر مرحله می سازد. در این مسئله، استراتژی بهینه شامل یک رویکرد جستجوی باینری است که ممکن است با ساختاری که ID3 می سازد همسو نباشد.

ID3 درخت های تصمیم را بر اساس حداکثر به دست آوردن اطلاعات در هر گره می سازد و تصمیمات آن بر اساس تقسیم ویژگی ها است. ماهیت مسئله ۹ توپ فلزی، با یک استراتژی مقایسه باینری، ممکن است برای ساختاری که ID3 به طور معمول ایجاد می کند، مناسب نباشد.

به طور خلاصه، در حالی که ID3 یک الگوریتم قدرتمند برای بسیاری از مسائل یادگیری ماشین است، ساختار و رویکرد آن ممکن است با ویژگی های خاص مسئله ۹ توپ فلزی همخوانی نداشته باشد، و ممکن است لزوماً درخت تصمیم گیری بهینه را برای این سناریو ایجاد نکند.

پاسخ ۳ - پرسش

۱-۳. قسمت ۱

Movie	Genre	Age rating	Language	Source	Film Location	Studio	Enjoyed
1	Drama	PG-13	French	Book Adaptation	Canda	Warner Bros	Yes
2	Drama	PG-13	English	Book Adaptation	Canda	Warner Bros	Yes
3	Horror	G	English	Book Adaptation	Canda	A24	No
4	Drama	PG-13	English	Book Adaptation	USA	A24	Yes

Genre = X_1 Age = X_2 language = X_3 source = X_4 location = X_5
 studio = X_6 Enjoyed = Y

$$P(X_1 = D) = \frac{3}{4}, \quad P(X_1 = H) = \frac{1}{4} \quad P(X_2 = PG-13) = \frac{3}{4}, \quad P(X_2 = G) = \frac{1}{4}$$

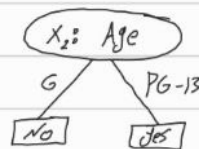
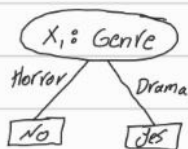
$$P(X_3 = F) = \frac{1}{4}, \quad P(X_3 = E) = \frac{3}{4} \quad P(X_4 = BA) = 1$$

$$P(X_5 = C) = \frac{3}{4}, \quad P(X_5 = U) = \frac{1}{4} \quad P(X_6 = W) = \frac{1}{2}, \quad P(X_6 = A) = \frac{1}{2}$$

$$P(Y = \text{yes} | X_1 = D) = 1, \quad P(Y = \text{no} | X_1 = H) = 0 \Rightarrow H(Y | X_1) = 0$$

$$P(Y = \text{yes} | X_2 = PG-13) = 1, \quad P(Y = \text{no} | X_2 = G) = 0 \Rightarrow H(Y | X_2) = 0$$

وقتی آنترופی ۰ می شود، یعنی ابهامی باقی نمی ماند و یادداشتن ژانر خروجی بصورت قطعی درست می آید.



از هر کدام از درخت های بالا را می توانم استفاده کرد.

۲-۳. قسمت ۲

Movie	Genre	Age rating	Language	Source	Film Location	Studio	Enjoyed
1	Drama	PG-13	French	Book Adaptation	Canda	Warner Bros	Yes
2	Drama	PG-13	English	Book Adaptation	Canda	Warner Bros	Yes
3	Horror	G	English	Book Adaptation	Canda	A24	No
4	Drama	PG-13	English	Book Adaptation	USA	A24	Yes
5	Drama	PG-13	French	Original Screenplay	Canda	Warner Bros	No

در صورت اضافه کردن یک ردیف داریم:

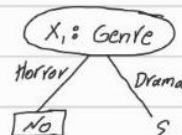
$$P(X_1 = D) = \frac{4}{5}, \quad P(X_1 = H) = \frac{1}{5} \quad P(X_2 = PG-13) = \frac{4}{5}, \quad P(X_2 = G) = \frac{1}{5}$$

$$P(X_3 = F) = \frac{2}{5}, \quad P(X_3 = E) = \frac{3}{5} \quad P(X_4 = BA) = \frac{4}{5}, \quad P(X_4 = OS) = \frac{1}{5}$$

$$P(X_5 = C) = \frac{4}{5}, \quad P(X_5 = U) = \frac{1}{5} \quad P(X_6 = W) = \frac{3}{5}, \quad P(X_6 = A) = \frac{2}{5}$$

$$H(Y|X_1) = 0.56 \checkmark \quad H(Y|X_2) = 0.56 \checkmark \quad H(Y|X_3) = 0.95 \quad H(Y|X_4) = 0.65 \quad H(Y|X_5) = 0.95$$

X_1 یا X_2 را می‌توانیم به عنوان ریشه انتخاب نمود که در این جا X_1 را انتخاب می‌کنیم.



دوباره مراحل را تکرار می‌کنیم.
ویزگی Genre و ردیف را در نظر نمی‌گیریم.

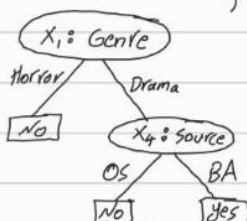
$$P(X_2 = PG-13) = 1$$

$$P(X_3 = F) = \frac{1}{2}, \quad P(X_3 = E) = \frac{1}{2} \quad P(X_4 = BA) = \frac{3}{4}, \quad P(X_4 = OS) = \frac{1}{4}$$

$$P(X_5 = C) = \frac{3}{4}, \quad P(X_5 = U) = \frac{1}{4} \quad P(X_6 = W) = \frac{3}{4}, \quad P(X_6 = A) = \frac{1}{4}$$

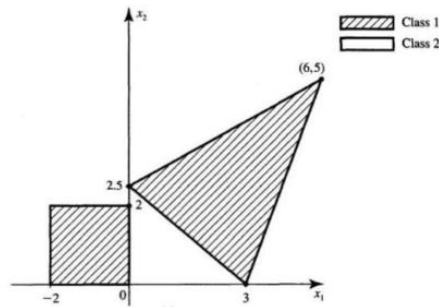
$$H(Y|X_2) = 0.81 \quad H(Y|X_3) = 0.5 \quad H(Y|X_4) = 0 \checkmark$$

چون به صفر رسیدیم نیازی به ادامه نیست و همین را به عنوان برگ انتخاب می‌کنیم.



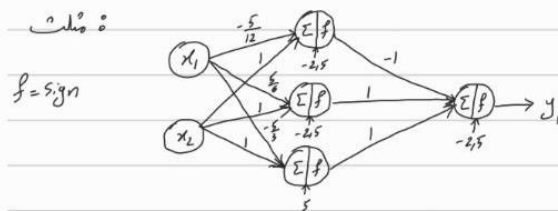
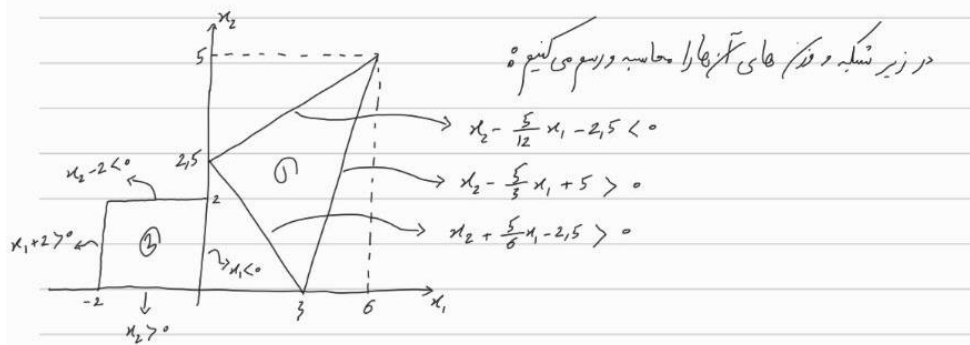
درخت نهایی به صورت زیر خواهد بود

پاسخ ۴ - پرسشی



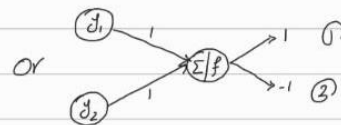
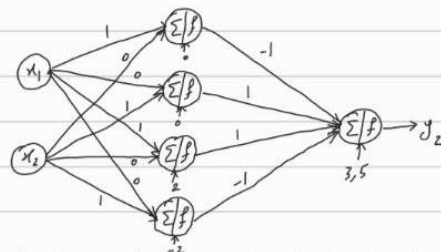
در این مسئله برای هر شکل یک شبکه داریم که در نهایت ابعام OR می شوند.

در شکل مثلث، 3 ضلع جداگانه داریم بنابراین به 3 نورون نیاز داریم و در شکل مربع نیز باید به این که 4 ضلع جدا کننده دارد بنابراین به 4 نورون نیاز است.



مربع:

$f = \text{sign}$



هر دو خروجی بالا در لایه دوم OR میشوند و خروجی نهایی را بدست میدهند. توجه کنید که برای ورودی x_1 و x_2 بین دو شبکه مشترک است و وزن های اتصال به شبکه گره های مخالف صفر است.

پاسخ ۵ - شبیه سازی

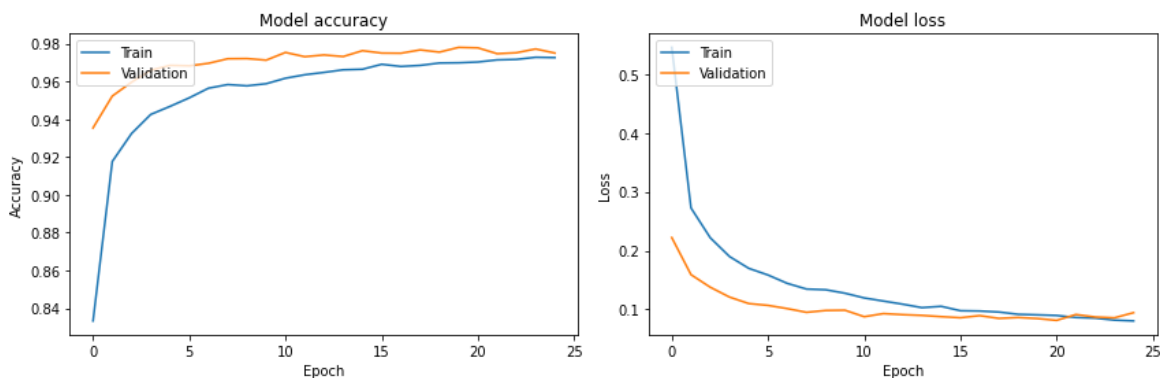
۵-۱. قسمت ۱ (MLP)

شبکه دارای سه لایه است: یک لایه مسطح، یک لایه متراکم با ۱۲۸ واحد و تابع فعال سازی ReLU، یک dropout (برای regularization)، یک لایه متراکم دیگر با ۶۴ واحد و تابع فعال سازی ReLU، و یک لایه متراکم نهایی با ۱۰ واحد و تابع فعال سازی softmax برای طبقه بندی وجود دارد.

```
Model: "sequential"
Layer (type)                Output Shape              Param #
=====
flatten (Flatten)           (None, 784)               0
dense (Dense)                (None, 128)              100480
dropout (Dropout)            (None, 128)               0
dense_1 (Dense)              (None, 64)                8256
dense_2 (Dense)              (None, 10)                650
=====
Total params: 109,386
Trainable params: 109,386
Non-trainable params: 0
```

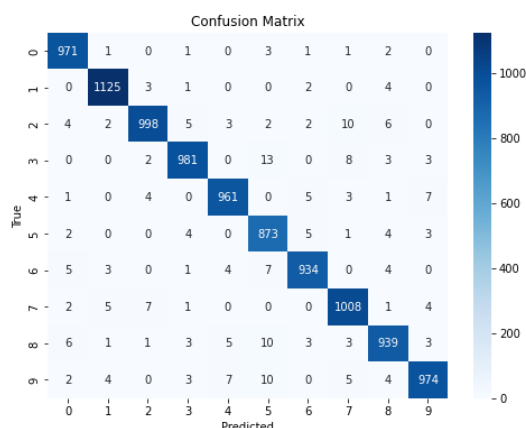
تعداد پارامترهای این مدل در تصویر بالا مشخص است.

نمودارهای منحنی یادگیری، خطای شبکه و دقت در طول آموزش بر روی داده های آموزش و ولیدیشن بصورت زیر رسم می شوند.



دقت در داده های تست بصورت زیر میباشد و یک ماتریس confusion برای تجزیه و تحلیل بیشتر ترسیم می شود.

```
313/313 [=====] - 1s 2ms/step - loss: 0.0885 - accuracy: 0.9764
Test Accuracy: 0.9764000177383423
```



بهترین زمان برای توقف آموزش شبکه به عملکرد validation بستگی دارد. می‌توان از توقف زودهنگام با نظارت بر مقدار loss استفاده کنید و زمانی که شروع به افزایش کرد، آموزش را متوقف کنید (که نشان دهنده overfit است). این را می‌توان با استفاده از یک callback در تابع `model.fit()` اضافه کرد.

۲-۵. قسمت ۲ (CNN)

در این مثال CNN:

شبکه دارای سه لایه کانولوشن با max-pooling و به دنبال آن دو لایه متراکم است که بصورت زیر می‌باشند.

```
Model: "sequential_1"
```

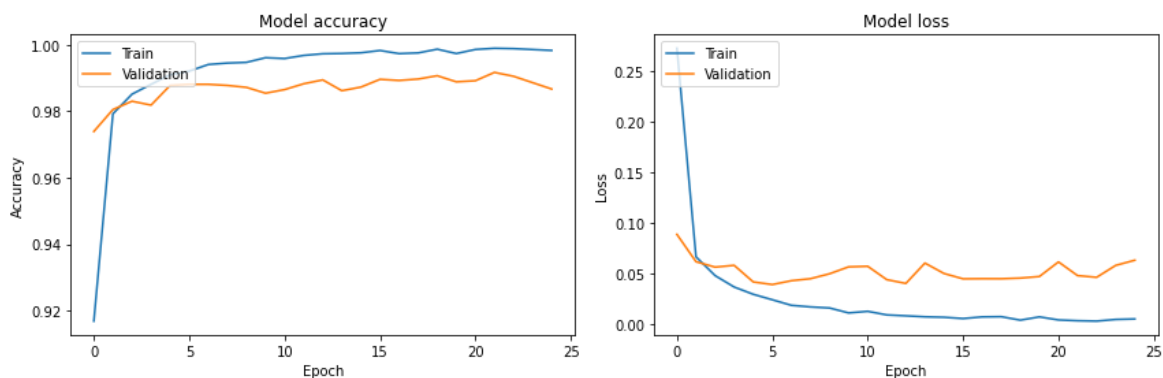
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
conv2d_2 (Conv2D)	(None, 3, 3, 64)	36928
flatten_1 (Flatten)	(None, 576)	0
dense_3 (Dense)	(None, 64)	36928
dense_4 (Dense)	(None, 10)	650

```

Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0

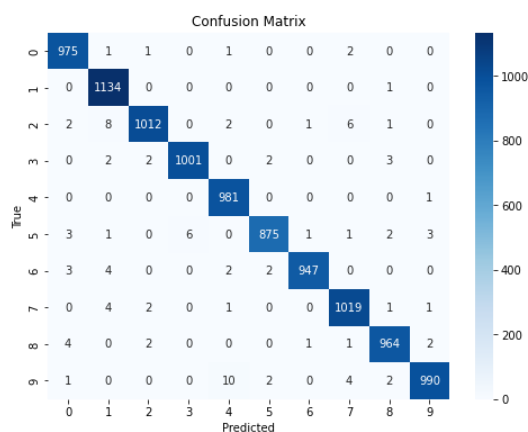
```

نمودارهای منحنی یادگیری، خطای شبکه و دقت در طول آموزش بر روی داده‌های آموزش و ولیدیشن بصورت زیر رسم می‌شوند.



دقت در داده های تست بصورت زیر میباشد و یک ماتریس confusion برای تجزیه و تحلیل بیشتر ترسیم می شود.

```
313/313 [=====] - 1s 3ms/step - loss: 0.0529 - accuracy: 0.9898
Test Accuracy: 0.989799976348877
```



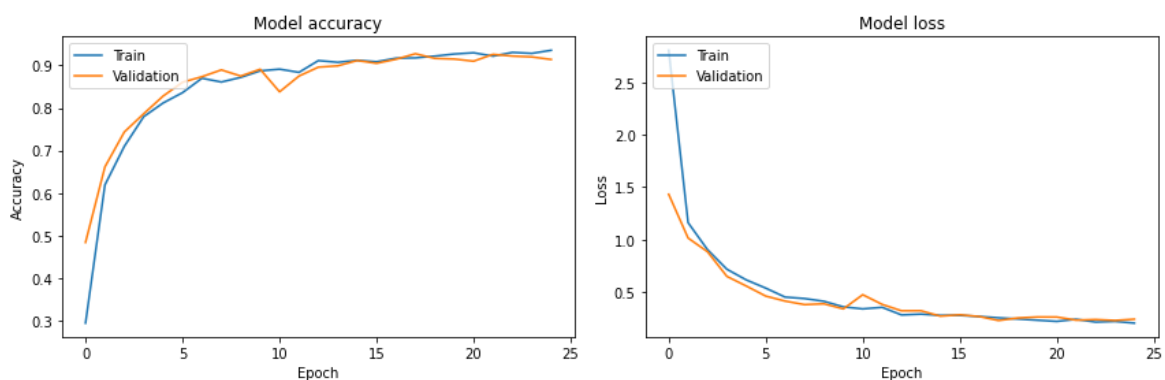
۳-۵. قسمت ۳ (Mobile Net)

MobileNet یک معماری شبکه عصبی کانولوشن با وزن سبک است که در اصل برای برنامه های موبایل و امبدد سیستم ها طراحی شده است. با این حال، استفاده از MobileNet برای انتقال یادگیری در مجموعه داده های MNIST استفاده میشود

ما از مدل MobileNetV2 به عنوان استخراج کننده ویژگی استفاده می کنیم و لایه های اضافی را برای کار طبقه بندی MNIST اضافه می کنیم.

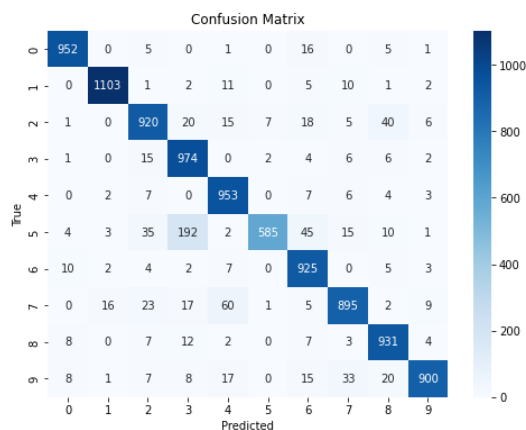
ما در اینجا به دلیل نبود فضای کافی و کمبود سخت افزار از قسمتی از داده استفاده نمودیم که این مقدار قابل تنظیم است.

نمودارهای منحنی یادگیری، خطای شبکه و دقت در طول آموزش بر روی داده های آموزش و ولیدیشن بصورت زیر رسم می شوند.



دقت در داده های تست بصورت زیر میباشد و یک ماتریس confusion برای تجزیه و تحلیل بیشتر ترسیم می شود.

313/313 [=====] - 5s 15ms/step - loss: 0.2612 - accuracy: 0.9138
Test Accuracy: 0.913800011444092



پاسخ ۶- شبیه سازی

اطلاعات دیتاست بصورت زیر قابل بیان است:

	0	1	2	3	4	5	6	7	8
count	10000	10000	10000	10000	10000	10000	10000	10000	10000
unique	2	7	16	7	14	6	5	2	40
top	<=50K	Private	HS-grad	Married-civ-spouse	Prof-specialty	Husband	White	Male	United-States
freq	7550	7379	3279	4651	1327	4104	8602	6784	9091

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 9 columns):
#   Column  Non-Null Count  Dtype
---  -
0     0      10000 non-null    object
1     1      10000 non-null    object
2     2      10000 non-null    object
3     3      10000 non-null    object
4     4      10000 non-null    object
5     5      10000 non-null    object
6     6      10000 non-null    object
7     7      10000 non-null    object
8     8      10000 non-null    object
dtypes: object(9)
memory usage: 703.2+ KB
```

که در آن ستون ۰، مقدار برچسب و بقیه ستون ها ویژگی ها می باشند.

الگوریتم ID3 را پیاده سازی کردیم و بر روی داده های آموزشی، آموزش دادیم. در نهایت مدل بدست آمده را با داده های تست و آموزش، تست کردیم و نتایج آن بصورت زیر میباشد.

Train data Accuracy is : 87.53999999999999

Test data Accuracy is : 81.23

در این الگوریتم در هر مرحله از Information gain برای انتخاب بهترین ویژگی استفاده شده است و در هنگام تست، در صورتی که پس از استفاده از تمام ویژگیها کلاس مشخص نشد، از رای گیری بین داده ها برای برچسب زنی برگها استفاده شده است و بیشترین تعداد نوع برچسب بعنوان برچسب برگ مشخص شده است.