به نام خدا

نام: فاطمه زهرا بخشنده **استاد: دکتر مرضیه داود آبادی**

شماره دانشجویی: 98522157 **درس مبانی یادگیری عمیق**

گزارش تمرین 1:

سوال اول:

الف) یک خروجی از 4 تا خروجی 1 است. پس:

$$p(1) = \frac{1}{4} = 0.25\,, \qquad p(0) = \frac{3}{4} = 0.75$$

در حالت عادی فرمول تئوری بیز را داریم:

$$p(y|x) = \frac{p(y,x)}{p(x)}$$

$$p(x) = \prod p(x^k)$$

هموارسازی لاپلاس یک تکنیک هموارسازی است که مشکل احتمال صفر را در Naïve Bayes حل می کند. با توجه به آن:

$$P(w'|positive) = \frac{\text{number of reviews with w' and y} = \text{positive} \; + \; \alpha}{\text{N} \; + \; \alpha * K}$$

α نشان دهنده پارامتر هموارسازی است.

K نشان دهنده تعداد ابعاد (ویژگی ها) در داده هااست.

N نشان دهنده تعداد review ها با y = positive است.

از این فرمول استفاده می کنیم که در واقع از همان فرمول Naïve Bayes گرفته شده است. α برابر 1 می گیریم.

$$p(a|0) = \frac{N(a,0)+1}{N(0)+(1*2)} = \frac{4+1}{6+2} = \frac{5}{8} \quad , \qquad p(b|0) = \frac{N(b,0)+1}{N(0)+(1*2)} = \frac{2+1}{6+2} = \frac{3}{8}$$

$$\rightarrow\ p(a|0) + p(b|0) = 1$$

$$p(a|1) = \frac{N(a,1)+1}{N(1)+(1*2)} = \frac{0+1}{2+2} = \frac{1}{4} \quad , \qquad p(b|1) = \frac{N(b,1)+1}{N(1)+(1*2)} = \frac{2+1}{2+2} = \frac{3}{4}$$

$$\rightarrow\ p(a|1) + p(b|1) = 1$$

ب)

$$p(C_i|x) = \frac{p(x,C_i)\,p(C_i)}{p(x)} = \frac{p(x,C_i)\,p(C_i)}{\sum_k^K p(x|C_k)p(C_k)}$$

choose Ci if $p(C_i|x) = \max_k p(C_k|x)$

برای aabaa:

$$p(0|aabaa) = \frac{p(aabaa|\,0)\,p(0)}{p(aabaa|0)p(0) + p(aabaa|1)p(1)}$$

$$p(aabaa|0)\,p(0) = p(a|0)\,p(a|0)\,p(b|0)\,p(a|0)\,p(a|0)\,p(0) = \frac{5}{8}*\frac{5}{8}*\frac{3}{8}*\frac{5}{8}*\frac{5}{8}*\frac{3}{4} \approx 0.043$$

$$p(aabaa|1)\,p(1) = p(a|1)\,p(a|1)\,p(b|1)\,p(a|1)p(a|1)p(1) = \frac{1}{4}*\frac{1}{4}*\frac{3}{4}*\frac{1}{4}*\frac{1}{4}*\frac{1}{4} \approx 0.0007$$

$$p(0|aabaa) = \frac{0.043}{0.043+0.0007} \approx 0.98\,, \qquad p(1|aabaa) = \frac{0.0007}{0.043+0.0007} \approx 0.02$$

پس با احتمال 98% عضو کلاس 0 است.

برای b:

$$p(b|0)\,p(0) = p(b|0)\,p(0) = \frac{3}{8}*\frac{3}{4} \approx 0.28$$

$$p(b|1)\,p(1) = p(b|1)\,p(1) = \frac{3}{4}*\frac{1}{4} \approx 0.18$$

$$p(0|b) = \frac{0.28}{0.28+0.18} \approx 0.61\,, \qquad p(1|b) = \frac{0.18}{0.28+0.18} \approx 0.39$$

پس با احتمال 61% عضو کلاس 0 است.

برای bba:

$$p(bba|0)\,p(0) = p(b|0)\,p(b|0)\,p(a|0)\,p(0) = \frac{3}{8} * \frac{3}{8} * \frac{5}{8} * \frac{3}{4} \approx 0.065$$

$$p(bba|1)\,p(1) = p(b|1)\,p(b|1)\,p(a|1)\,p(1) = \frac{3}{4} * \frac{3}{4} * \frac{1}{4} * \frac{1}{4} \approx 0.035$$

$$p(0|bba) = \frac{0.065}{0.065 + 0.035} \approx 0.65\,, \qquad p(1|bba) = \frac{0.035}{0.065 + 0.035} \approx 0.35$$

پس با احتمال 65% عضو کلاس 0 است.

برای bbbb:

$$p(bbbb|0)\,p(0) = p(b|0)\,p(b|0)\,p(b|0)\,p(b|0)\,p(0) = \frac{3}{8} * \frac{3}{8} * \frac{3}{8} * \frac{3}{8} * \frac{3}{4} \approx 0.015$$

$$p(bbbb|1)\,p(1) = p(b|1)\,p(b|1)\,p(b|1)\,p(b|1)\,p(1) = \frac{3}{4} * \frac{3}{4} * \frac{3}{4} * \frac{3}{4} * \frac{1}{4} \approx 0.079$$

$$p(0|bbbb) = \frac{0.015}{0.015 + 0.079} \approx 0.16\,, \qquad p(1|bbbb) = \frac{0.079}{0.015 + 0.079} \approx 0.84$$

پس با احتمال 84% عضو کلاس 1 است.

Data	Class
aabaa	0
b	0
bba	0
bbbb	1

منابع: لینک و اسلاید ها

سوال دوم:

کدهای نوت بوک هارا اجرا می کنیم. اورورهایی در آن ها وجود دارد. مثلا در basic به ارور های زیر برمی‌خوریم که باید به آن ها توجه کنیم.

```
'tuple' object does not support item assignment

KeyError: 2    (in dictionary)

TypeError: 'int' object is not iterable
```

ارور های نوت بوک numpy که مربوط به shape بودند. (در هنگام عملگر dot و کم کردن دو ماتریس):

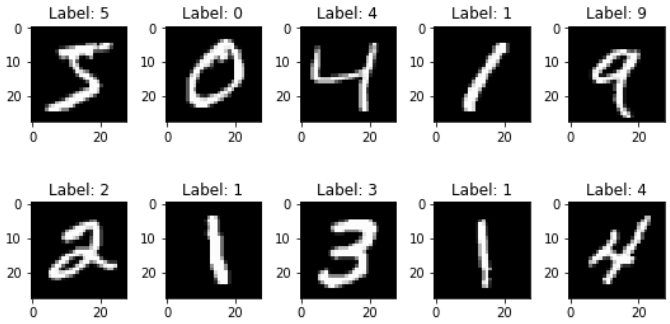
```
ValueError: shapes (3,4) and (3,4) not aligned: 4 (dim 1) != 3 (dim 0)

ValueError: operands could not be broadcast together with shapes (3,4) (3,)
```

ارور موجود در pytorch: در cell آخر مربوط به استفاده از CUDA، CPU را در colab به GPU تبدیل می کنیم.

سوال سوم:

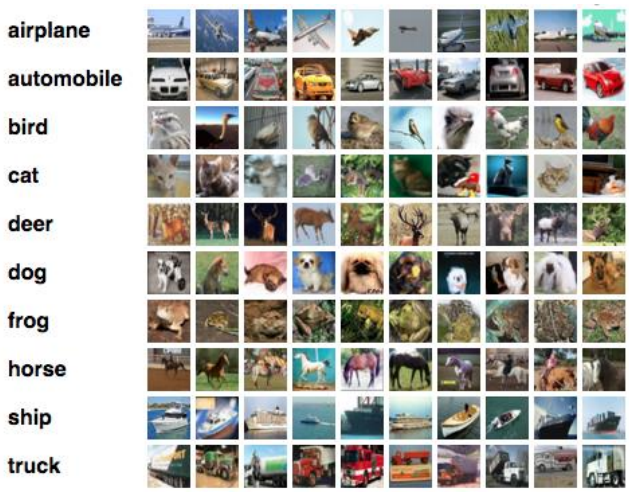
الف)

1- Mnist: دیتاست Mnist در سال 1998 معرفی شد و به یک معیار استاندارد برای تسک های classification تبدیل شد. این dataset شامل دو بخش است. 60000 تصویر train و 10000 تصویر test که همگی grayscale می باشند. تصاویر شامل اعداد دست نویس با شماره 0 تا 9 هستند. پس 10 کلاس مختلف داریم و هر تصویر یک label دارد که می تواند از 0 تا 9 باشد. (10 مقدار مختلف).



CIFAR-10: دیتاست CIFAR-10 مجموعه ای از تصاویر مختلف است و یک مجموعه داده بسیار پایه و محبوب برای تمرین یادگیری ماشین و بینایی کامپیوتر است. این dataset شامل 60000 تصویر 32 در 32 است. و شامل دو بخش است. 50000 تصویر train و 10000 تصویر test که همگی RGB می باشند. پس

ابعاد هر تصویر (32, 32, 3) است. تصاویر در 10 کلاس مختلف قرار می گیرند و 6000 تصویر از هر کلاس وجود دارد. هر تصویر یک label دارد که که کلاس آن را مشخص می کند و می تواند از 0 تا 9 باشد.



FER-2013: دیتاست FER-2013 در کنفرانس بین المللی یادگیری ماشین در سال 2013 منتشر شد. این dataset شامل تصاویر 48 در 48 از چهره ها است. و شامل دو بخش است. 28709 تصویر train و 3589 تصویر تست public و 3589 تصویر تست private دارد و روی هم 7178 تا تصویر تست دارد. که همگی grayscale می باشند. چهره ها به طور خودکار ثبت شده اند به طوری که صورت کم و بیش در مرکز قرار گرفته و تقریباً یک مقدار فضای مشخص را در هر تصویر را اشغال می کند. هر تصویر در یکی از 7 کلاس (0 = عصبانی، 1 = انزجار، 2 = ترس، 3 = خوشحال، 4 = غمگین، 5 = تعجب، 6 = خنثی) قرار می گیرد پس 7 label مختلف داریم. Task مربوط به آن، دسته بنده هر چهره در این 7 دسته، بر اساس احساسات نشان داده شده در حالت آن چهره است.

2- وقتی دیتاست هارا لود می کنیم، y_train یک rank 1 array است. که باید آن را به فرم categorical تبدیل کنیم. با استفاده از متد ()to_categorical، یک آرایه numpy یا یک vector که دارای اعداد صحیح است که دسته‌های مختلف را نشان می‌دهد، می‌تواند به یک آرایه numpy یا ماتریسی تبدیل شود که دارای مقادیر باینری است و دارای تعداد سطر برابر طول vector ورودی و تعداد ستون برابر با تعداد کلاس ها است. اگر کلاس‌ها در یک ماتریس باینری توزیع نشده باشند، یک مدل classification چند کلاسه خوب

کار نمی‌کند. چون در این دیتاست ها برای هر دو لیبل مختلف اشتباه باید خطا برابر باشد که در حالت integer این قضیه در نظر گرفته نمی شود. مثلا وزن عدد 3، سه برابر وزن عدد 1 است و شبکه اهمیت آن را بیشتر در نظر می گیرد. اما باید مقدار باینری 0 یا 1 بودن هر کلاس را در لیبل مورد نظر، در نظر بگیریم (one hot encoding). دلیل دیگر آن هم تابع activation لایه آخر است، برای اینکه به صورت احتمال بدست بیایند و کاملا گسسته نباشند. پس از این متد استفاده می کنیم تا وزن اعداد روی شبکه تاثیر نداسته باشند و لیبل های داده های آموزشی خود را قبل از اینکه به مدل خود منتقل کنیم، تغییر دهیم.

3- در دیتاست Mnist، x_train دیتای آموزشی میباشد که دارای شکل (60000, 28, 28) است. بعد اول تعداد داده های آموزشی که 60000 تاست، بعد دوم و سوم هم ابعاد تصویر یعنی width و height تصاویر هستند که 28*28 هستند. چون تصاویر grayscale هستند سه بعدی است. y_train لیبل های دیتای آموزشی هستند که دارای شکل (60000, 10) است. بعد اول تعداد لیبل ها که 60000 و بعد دوم تعداد کلاس های مختلف است. در واقع چون از to_categorical استفاده کردیم بعد دوم آن 10 شده که به صورت one hot است.

در دیتاست CIFAR-10، x_train دارای شکل (50000, 32, 32, 3) است. بعد اول تعداد داده های آموزشی که 50000 تاست، بعد دوم و سوم هم ابعاد تصویر یعنی width و height تصاویر هستند که 32*32 هستند. چون تصاویر RGB هستند بعد چهارم هم سه پارامتر دارد. y_train دارای شکل (50000, 10) است. بعد اول تعداد لیبل ها که 50000 و بعد دوم تعداد کلاس های مختلف است که 10 تاست.

دیتاست FER-2013 هم برای x_train، بعد اول 28709 تا دیتا و بعد دوم و سوم ابعاد تصاویر که 48 در 48 است و بعد چهارم عدد 1 که یعنی یک کاناله است، چون grayscale هستند. y_train هم بعد اول 28709 تا لیبل و بعد دوم 7 تا کلاس میباشد. (اگر برای یک batch را در نظر بگیریم بعد اول هردو 64 میشود.)

4- از imageDataGenerator می توان برای Data Augmentation استفاده کرد. این کلاس از تصاویر موجود در دیتاست کپی می گیرد و با گرفتن پارامتر های مختلف مثل zoom_range، vertical_flip، horizontal_flip و .. روی تصویر تغییراتی مثل zoom، flip، rotation، shift انجام می دهد. این کار باعث می شود مدل قویتر و پایدارتر شود و بدون اضافه کردن دیتا توسط خودمان، دیتای بیشتری را پوشش میدهد. با این کار مشکل کمبود دیتا تا حدی برطرف می شود.

همچنین با استفاده imageDataGenerator ابتدا دیتاست را لود می کنیم. و سپس آن را به صورت دسته دسته (batch) به مدل وارد می کنیم. در واقع همه تصاویر را باهم به مدل نمی دهیم که باعث میشود حافظه زیادی ذخیره شود.

منابع:

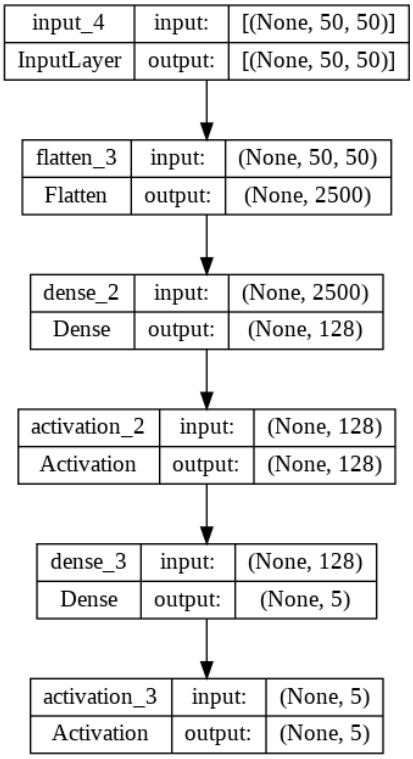
https://www.kaggle.com/

https://medium.com/

https://www.geeksforgeeks.org/

https://stackoverflow.com/questions/44110426/when-to-use-to-categorical-in-keras#:~:text=You%20use%20to_categorical%20to%20transform,a%20classification%20model%20without%20that.

ب) خروجی رسم دو مدل:



Summary مدل اول:

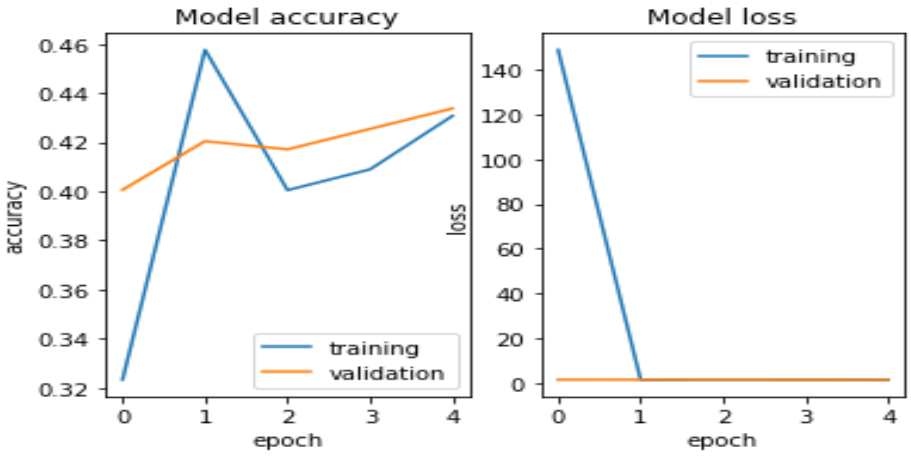
```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 2500)              0

 dense (Dense)               (None, 128)               320128

 activation (Activation)     (None, 128)               0

 dense_1 (Dense)             (None, 5)                 645

 activation_1 (Activation)   (None, 5)                 0

=================================================================
Total params: 320,773
Trainable params: 320,773
Non-trainable params: 0
```

Summary مدل دوم:

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 50, 50)]          0

 flatten_3 (Flatten)         (None, 2500)              0

 dense_2 (Dense)             (None, 128)               320128

 activation_2 (Activation)   (None, 128)               0

 dense_3 (Dense)             (None, 5)                 645

 activation_3 (Activation)   (None, 5)                 0

=================================================================
Total params: 320,773
Trainable params: 320,773
Non-trainable params: 0
```

در summary ساختار لایه های مختلف مدل و جزئیات هر لایه مثل نام و نوع لایه، shape خروجی، activation استفاده شده و تعداد نورون ها و پارامترهای موجود در هر لایه را می نویسد. همچنین تعداد همه پارامتر ها و پارامتر های آموزش پذیر مدل را نیز مشخص می کند.

د) مدل دیتاست Mnist:



در epoch اول دقت یکهو افزایش یافته، در epoch دوم یکدفعه کاهش یافته و در epoch سوم و چهارم به طور متعادل تری افزایش می یابد. در کل دقت مدل از ابتدا تا انتها طی فرایند یادگیری برای train و val افزایش یافته است و تقریبا برابر است. اما می توانستیم مدل بهتری انتخاب کرده یا زمان بیشتری train کنیم و افزایش دقت متعادلتر و بهتری داشته باشیم.

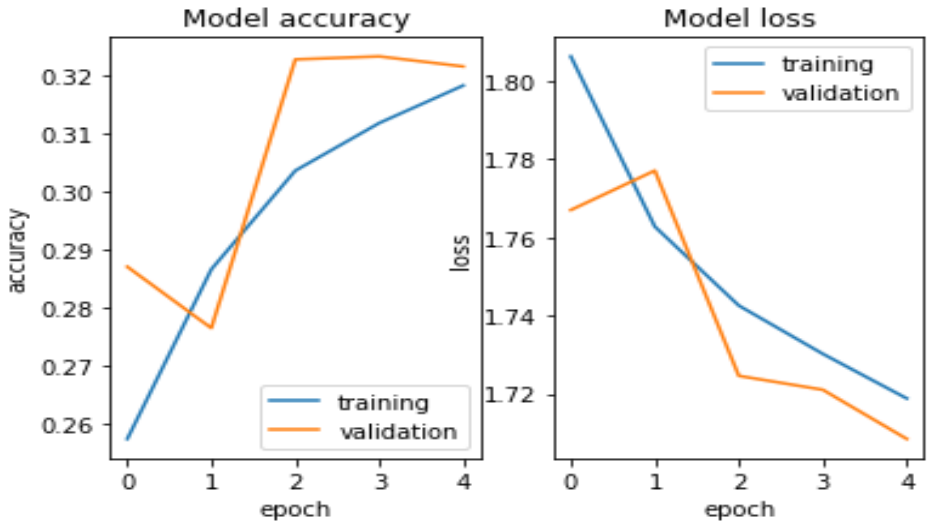
خروجی تابع evaluate نشان می دهد که میزان دقت مدل روی داده آزمون 0.4378 و میزان خطای آن 1.4860 بوده است.

```
157/157 [==============================] - 0s 3ms/step - loss: 1.4860 - accuracy: 0.4378
[1.4860379695892334, 0.43779999017715454]
```

پیشبینی مدل روی 3 نمونه داده آزمون که دو پیشبینی از سه تا درست بوده اند:

```
1/1 [==============================] - 0s 325ms/step
y_true : 5, y_predicted : 6
1/1 [==============================] - 0s 51ms/step
y_true : 6, y_predicted : 6
1/1 [==============================] - 0s 81ms/step
y_true : 2, y_predicted : 2
```

مدل دیتاست FER-2013:



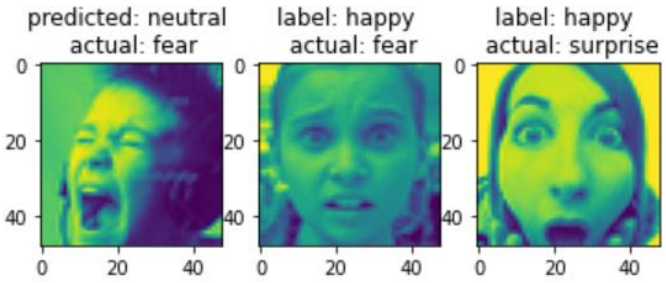
برای داده val در epoch اول کاهش دقت داریم. سپس در epoch دوم یکهو افزایش دقت داشته و در epoch سوم و چهارم به آرامی کمی کاهش دقت داشته ایم. میزان دقت روی داده train به طور متعادل همواره در حال افزایش است. در انتها دو دقت به هم میل کرده اند و تقریبا برابرند.

خروجی تابع evaluate نشان می دهد که میزان دقت مدل روی داده آزمون 0.3431 و میزان خطای آن 1.6826 بوده است.

```
113/113 [==============================] - 3s 24ms/step - loss: 1.6826 - accuracy: 0.3431
[1.6825716495513916, 0.34313178062438965]
```

خروجی پیشبینی چند نمونه از داده های آزمون:

```
113/113 [==============================] - 3s 23ms/step
Text(0.5, 1.0, 'label: happy \n actual: surprise')
<Figure size 432x288 with 0 Axes>
```



نه، مطابقت ندارند. در واقع دقت خیلی پایینی داریم و بسیاری از داده های تست اشتباه پیشبینی می‌شوند. Bias خیلی بالایی داریم. این موضوع دلایل مختلفی دارد.

با توجه به اینکه این دیتاست تصاویر چهره های انسان هارا دارا هست، نیاز به شبکه ای داریم که تمام حالات پیچیده آن را شناسایی کند. پس نیاز به تعداد نورون های بیشتر و لایه های بیشتری داریم تا شبکه بزرگتری بسازیم و در هر لایه یک سری فیچرهای پیچیده تر را از تصویر در بیاورد. همچنین باید زمان training را افزایش دهیم تا مدل بهتر آموزش ببیند. تعداد epoch ها در حال حاظر خیلی کم است.

پس از این ها استفاده از optimizer قوی تر و تغییر hyperparameter ها مانند learning_rate و batch_size و تعداد hidden units هر لایه و ... ممکن است روی بهینه شدن یادگیری تاثیر خوبی بگذارند. که بهترین مقدار این hyperparameter هارا برای مدلمان باید با مقایسه و تجربه بدست بیاوریم.

اگر پس از این ها واریانس ما بالا رفت باید دیتاهای را اضافه کنیم و regularization انجام دهیم که می‌تواند data augmentation باشد.