

به نام خدا



تمرین سری پنجم درس یادگیری عمیق

دکتر محمدی

محمد یارمقدم

۹۶۴۶۲۱۰۴

سوال اول)

الف) می توان با این مسئله به عنوان یک مسئله **localization** نگاه کرد. پس به ازای هر نقطه، دو نورون (یکی برای X و یکی برای Y) در نظر میگیریم که در مجموع ۱۰ نورون در لایه آخر خواهیم داشت که تابع فعالسازی آن ها **sigmoid** می باشد. تابع ضرر مناسب برای این مسئله با توجه به فضای پیوسته آن **MSE** خواهد بود. زیرا اعداد باید بسیار نزدیک باشند و محدوده خاصی ندارند.

```
def get_model(IMAGE_SHAPE):
    with tf.device("/cpu:0"):
        # Define model
        X = Input(shape=IMAGE_SHAPE)
        baseModel = ResNet50(include_top=False)
        pooled = GlobalAveragePooling2D()(baseModel(X))
        dense = Dense(2 * NUM_LANDMARKS, activation="sigmoid", kernel_initializer="glorot_uniform")
        out = dense(pooled)
        model = Model(inputs=X, outputs=out)
    return model
```

برای لایه آخر از ۱۵۲ نورون (مختصات **landmark** ها چهره را در تصویر مشخص می کند) استفاده شده است.

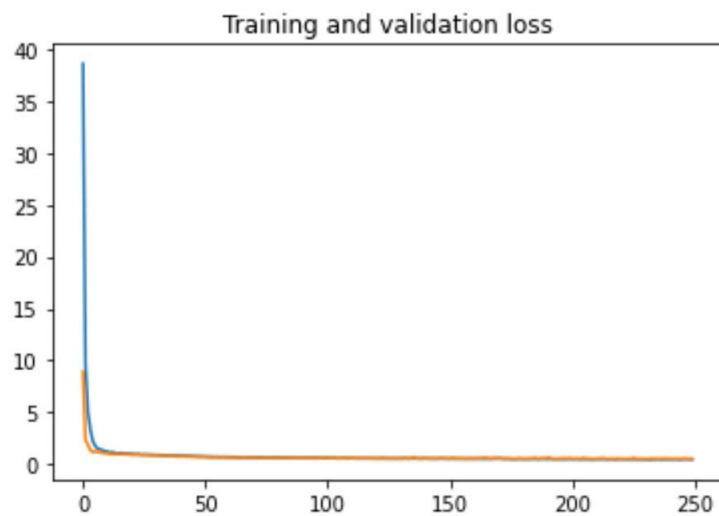
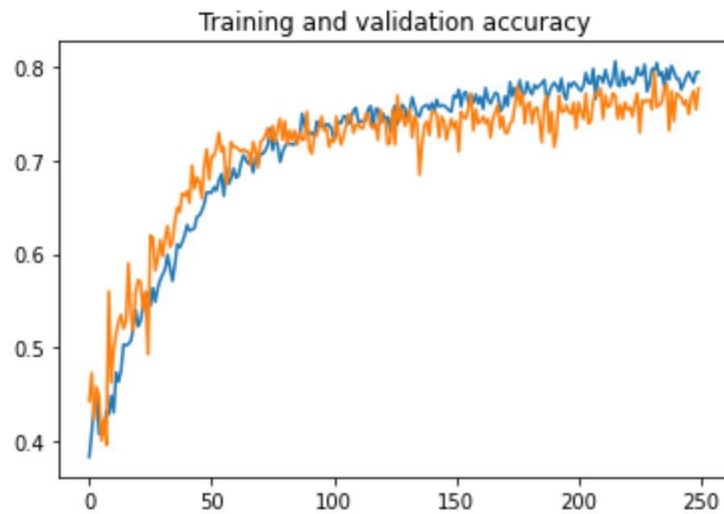
ب) این کد از تابع فعالسازی **sigmoid** و تابع ضرر **MSE** استفاده می کند. در واقع برای تابع ضرر زیر تعریف شده است که اگر مقادیری وجود نداشته باشد مقدار **loss** آنها صفر در نظر گرفته میشود.

```
def mse_with_dontcare(T, Y):
    ##Find the pairs in T having x-y coordinate = (0, 0)
    #Reshape to have x-y coordinate dimension
    T_resaped = tf.reshape(T, shape=[-1, 2, NUM_LANDMARKS])
    Y_resaped = tf.reshape(Y, shape=[-1, 2, NUM_LANDMARKS])
    #Calculate Euclidean distance between each pair of points
    distance = tf.norm(T_resaped - Y_resaped, ord='euclidean', axis=1)

    #If summing x and y yields zero, then (x, y) == (0, 0)
    T_summed = tf.reduce_sum(T_resaped, axis=1)
    zero = tf.constant(0.0)
    mask = tf.not_equal(T_summed, zero)
    #Get the interested samples and calculate loss with Mean of Euclidean distance
    masked_distance = tf.boolean_mask(distance, mask)
    return tf.reduce_mean(masked_distance)
```

سوال دوم)

برای حل این سوال از ۴ نورون در لایه آخر با تابع فعالسازی softmax و تابع ضرر categorical_crossentropy استفاده شد مه به دقت ۷۹,۵ روی داده های train و دقت ۷۷,۷۴ روی داده های validation انجامید.



سوال سوم)

در این سوال هدف بررسی توابع فعالسازی و ضرر در داده های خبری است. برای این کار ابتدا یک شبکه ساده با ۳ لایه گفته شده را تشکیل داده و سپس در آرگومان های تابع `compile` تابع ضرر و در فعالساز لایه آخر توابع مختلف را تست و نمودار هر بخش را از نتایج رسم می کنیم.

```
loss 1: 0.0366843082010746 accuracy 1: 0.7217274904251099
loss 2: 0.920823335647583 accuracy 2: 0.784060537815094
loss 3: 0.011005123145878315 accuracy 3: 0.5445235967636108
loss 4: 0.007502493914216757 accuracy 4: 0.74799644947052
loss 5: 0.010472523048520088 accuracy 5: 0.6308993697166443
```

پس از مشاهده نتایج تست هر ۵ حالت خواسته شده در سوال، دریافت میشود که خطای روش دوم با اختلاف فاحش بالاتر از سایر موارد است پس بدترین روش این مورد محسوب می شود زیرا روند آموزش را به خوبی طی نکرده است.

اما روش چهارم هم دقت قابل قبولی در میان روش ها دارد هم خطای بسیار کمی دارد پس بهترین روش این روش محسوب میشود.

پس بدون دیدن دقت و خطای اعتبارسنجی و صرف دقت و خطای داده های تست این نتیجه بدست می آید.

اما با دیدن نتایج اعتبار سنجی نتایج متفاوت است. چون در حالت چهارم با بالا رفتن تعداد epoch ها اورفیت رخ داده و خطوط نمودار های خطا و دقت از یکدیگر فاصله میگیرد پس نتیجه می شود که فقط داده اصلی در حال یادگیری است و روی همین داده ها کار خواهد کرد. به علاوه ترکیب این دو با هم برای رگرشن داده های صفر و یک استفاده میشود.

در روش سوم نیز آموزش دقت بالایی نمی یابد و روند ثابت به خود میگیرد که مناسب نیست.

بهترین روش، روش دوم است که ترکیب `categorical_crossentropy` با `softmax` است که روش دوم است. این ترکیب برای داده های `multilabel` استفاده میشود پس بهترین نتیجه را میدهد.

پس بهترین روش دوم و بدترین به ترتیب روش چهارم و سوم میشود.

