

تمرین سوم

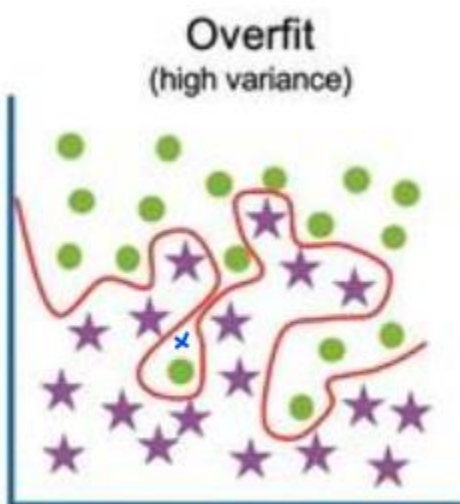
سینا اسکندری ۹۷۵۲۱۰۵۴

۱-

(الف)

Overfit به معنای این است که شبکه قابلیت تعمیم سازی روی داده هایی که تا الان ندیده، ندارد. زمانی متوجه overfit می شویم که دقت شبکه روی داده train زیاد است مثلاً ۹۹ درصد و دقت داده test بسیار کمتر از آن است مثلاً ۷۵ درصد. برای جلوگیری از آن می توان از داده بیشتر استفاده کرد یا از regularization و dropout استفاده کرد.

(ب)



با توجه به نمودار اگر نقطه ای در جای مشخص شده قرار بگیرد احتمال اینکه به دسته بنفش تعلق داشته باشد بیشتر است ولی مدل ما سبز پیش بینی می کند.

$$x=1, y=2.0$$

$$\text{Feed forward} \rightarrow \sum w_n + b = 1 + 1 + 1 - 1 - 1 = 1 \rightarrow \hat{y} = 1$$

$$J = (2.0 - 1)^2 = 1.0$$

$$\frac{\partial J}{\partial w} = \frac{\partial}{\partial w} (y - wx)^2 = 2(y - wx)(-x) = 2(2.0 - 1)(-1) = -2.0$$

$$w \leftarrow (1 - \epsilon \lambda)w - \epsilon \frac{\partial J}{\partial w}$$

$$\Rightarrow w_1 = (1 - 0.09) \times 1 - (0.1)(-2.0) = 1.1$$

$$w_2 = 0.91 \times 2 + 2.0 = 3.82$$

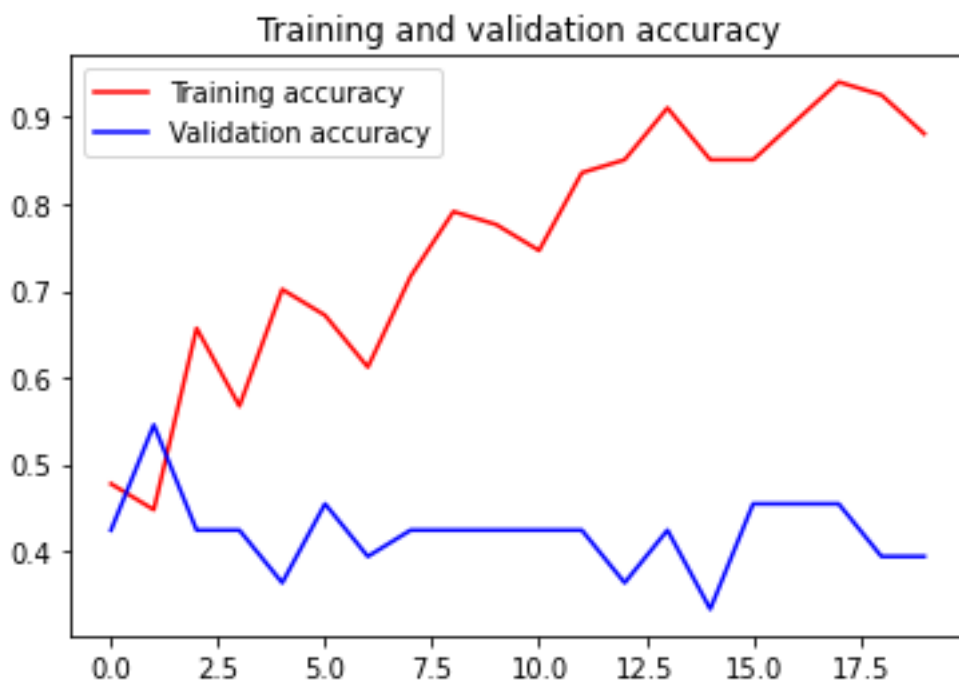
$$w_3 = 0.91 \times 1 + 2.0 = 3.91$$

$$w_4 = 0.91 \times (-2) + 2.0 = 1.82$$

$$w_5 = 0.91 \times (-1) + 2.0 = 1.89$$

-۳

طبق مراحل خواسته شده در نوتبوک پیش میرویم که نتایج و پیاده سازی مشخص است. در حالت بدون augmentation دقت به این صورت است



```
# Evaluate your model on the test data
```

```
#####
```

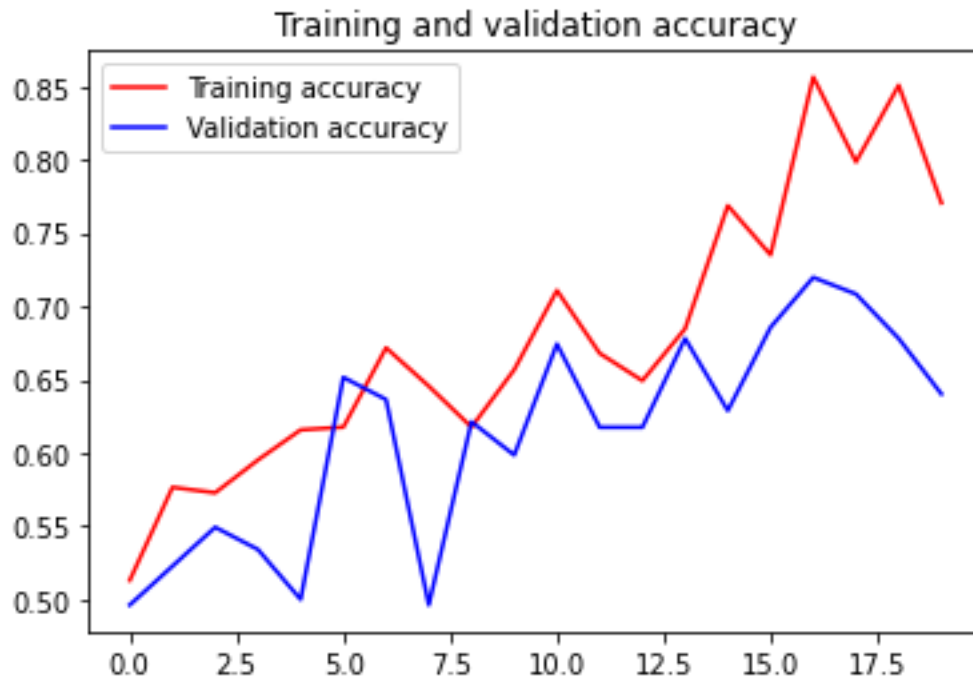
```
## PUT YOUR CODE HERE ##
```

```
#####
```

```
model.evaluate(X_test, y_test)
```

```
2/2 [=====] - 0s 11ms/step - loss: 1.6322 - accuracy: 0.3939  
[1.632162094116211, 0.39393940567970276]
```

در حالت اعمال augmentation:



```
# Evaluate your model on the test data
```

```
#####
```

```
## PUT YOUR CODE HERE ##
```

```
#####
```

```
model2.evaluate(X_test, y_test)
```

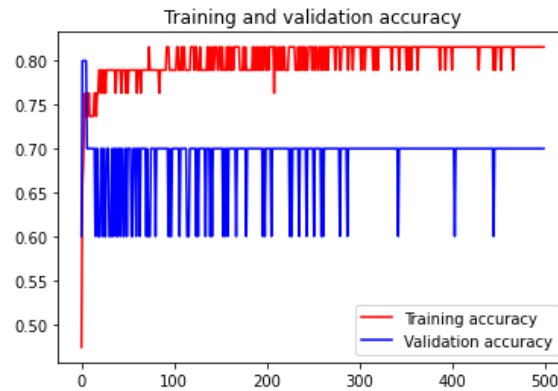
```
9/9 [=====] - 0s 3ms/step - loss: 3.4295 - accuracy: 0.6402  
[3.4295058250427246, 0.6401515007019043]
```

مشخص است که دقت train کم شده است ولی دقت validation و test نسبتاً بهتر شده است که تا حدی overfitting کمتر شده است ولی کامل از بین نرفته است.

۴-

برای جلوگیری از overfitting از regularization، dropout و early stopping استفاده می‌کنم که پیاده‌سازی و نتایج در نوت‌بوک مشخص است. که در اینجا regularization از early stopping و از dropout به ترتیب بهتر عمل کرده است.

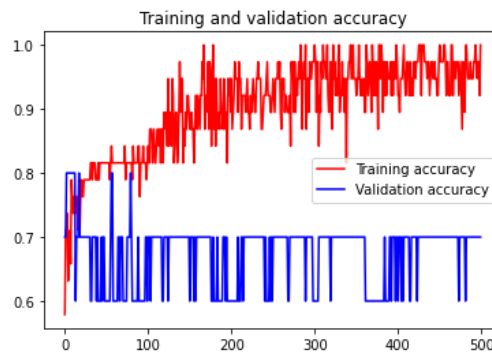
:Regularization



```
model2.evaluate(X_test, y_test)
```

```
1/1 [=====] - 0s 119ms/step - loss: 0.6787 - accuracy: 0.8333  
[0.6786808967590332, 0.8333333134651184]
```

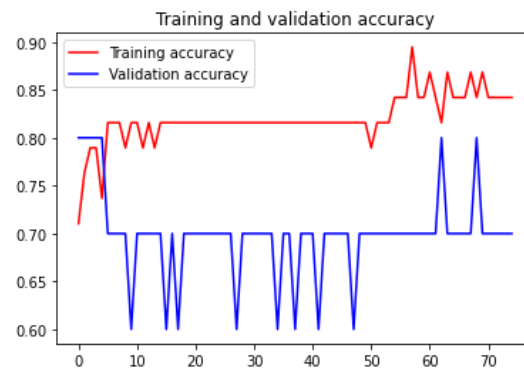
:Dropout



```
[46] model3.evaluate(X_test, y_test)
```

```
1/1 [=====] - 0s 92ms/step - loss: 8.7425 - accuracy: 0.6667  
[8.74245834350586, 0.6666666865348816]
```

:Early stopping



```
model4.evaluate(X_test, y_test)
```

```
1/1 [=====] - 0s 105ms/step - loss: 2.4701 - accuracy: 0.8333  
[2.470087766647339, 0.8333333134651184]
```