

تمرین شماره ۲

سینا اسکندری ۹۷۵۲۱۰۵۴

۱-

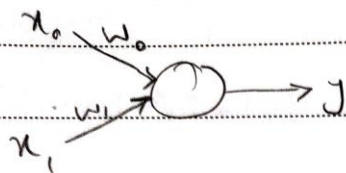
(الف)

با توجه به نمودار همه optimizer ها در iteration ها اولیه تقریباً عین هم عمل می کنند و در مراحل آخر است که تفاوت بسیار حاصل می شود که adam که ویژگی های مثبت Adagrad و RMSProp را دارد از همه بهتر عمل کرده است بعد از آن نیز SGD + Nesterov عملکرد خوبی داشته است.

(ب)

به طور کلی adam از بقیه بهتر عمل می کند ولی مثلاً اگر گرادیان های پراکنده داشته باشیم adagrad گزینه خوبی است.

<https://datascience.stackexchange.com/questions/10523/guidelines-for-selecting-an-optimizer-for-training-neural-networks>



epoch 1:

$$\text{data 1: } z = \sum w_i x_i + b = 4 - 1 + 1 = 4 \rightarrow \hat{y} = \sigma(z) = 0.99$$

$$J = \frac{1}{2} (y - \hat{y})^2 \rightarrow \frac{\partial J}{\partial \hat{y}} = -(y - \hat{y}) = \hat{y} - y = -0.01$$

$$\frac{\partial \hat{y}}{\partial z} = \frac{\partial}{\partial z} (\sigma(z)) = (1 - \sigma(z)) \sigma(z) = 0.01 \times 0.99 = 0.0099 \approx 0.01$$

$$\frac{\partial z}{\partial w_0} = x_0 = 4, \quad \frac{\partial z}{\partial w_1} = x_1 = -1, \quad \frac{\partial z}{\partial b} = 1$$

$$\rightarrow \frac{\partial J}{\partial w_0} = -0.01 \times 0.01 \times 4 = -0.0004$$

$$\frac{\partial J}{\partial w_1} = -0.01 \times 0.01 \times (-1) + 0.0001, \quad \frac{\partial z}{\partial b} = -0.01 \times 0.01 = -0.0001$$

نیز باید / فرجه

$$\rightarrow w_0 = 4 - (1)(-0.0004) = 4.0004$$

$$w_1 = -1 - (1)(-0.0001) = -0.9999$$

$$b = 1.0001$$

data 1: $w_0 = 1,001$, $w_1 = 1,001$, $b = 1,001$

$z = 1,001 - 1,001 + 1,001 = 1,001 \rightarrow \sigma(z) = 0,99$

$$\frac{\partial J}{\partial \hat{y}} = 0,99$$

$$\frac{\partial \hat{y}}{\partial z} = 0,99 \approx 0,1$$

$$\frac{\partial z}{\partial w_0} = 1$$

$$\frac{\partial z}{\partial w_1} = -1$$

$$\frac{\partial z}{\partial b} = 1$$

$$\frac{\partial J}{\partial w_0} = 0,99 \times 0,1 \times 1 = 0,099 \approx 0,1, \quad \frac{\partial J}{\partial w_1} = -0,1$$

$$\frac{\partial J}{\partial b} = 0,1 \rightarrow w_0 = 1,001 - 0,1 = 1,901$$

$$w_1 = 1,001 + 0,1 = 1,101$$

$$b = 1,001 - 0,1 = 1,901$$

epoch 2:

data 1: $z = 0,109 - 1,101 + 1,901 = 0,909 \rightarrow \sigma(z) = 0,99$

$$\frac{\partial J}{\partial \hat{y}} = 0,01$$

$$\frac{\partial \hat{y}}{\partial z} = 0,1$$

$$\frac{\partial z}{\partial w_0} = 1$$

$$\frac{\partial z}{\partial w_1} = -1$$

$$\frac{\partial J}{\partial w_0} = -0,001$$

$$\frac{\partial J}{\partial b} = -0,001$$

$$w_0 = 1,901 + 0,001 = 1,902$$

$$\rightarrow w_1 = 1,101 - 0,001 = 1,1$$

$$\frac{\partial J}{\partial w_1} = 0,001$$

$$b = 1,901 + 0,001 = 1,902$$

data ۲:

$$z = 1,909 - 2,4 + 1,902 = 1,401 \rightarrow \sigma(z) = 0,18$$

$$\frac{\partial J}{\partial \hat{y}} = 0,18 \quad \frac{\partial \hat{y}}{\partial z} = 0,14$$

$$\rightarrow \frac{\partial J}{\partial w_0} = 0,14 \times 0,18 = 0,0252 \quad \frac{\partial J}{\partial w_1} = -2 \times 0,14 \times 0,18 = -0,1008$$

$$\frac{\partial J}{\partial b} = 0,18$$

$$\rightarrow w_0 = 1,909 - 0,18 = 1,729$$

$$w_1 = 1,4 + 0,1008 = 1,5008$$

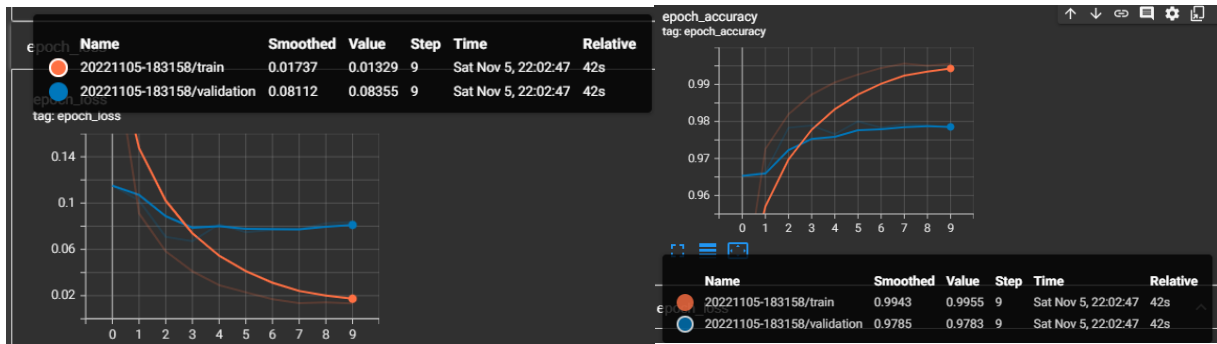
$$b = 1,98 - 0,18 = 1,8$$

-۳

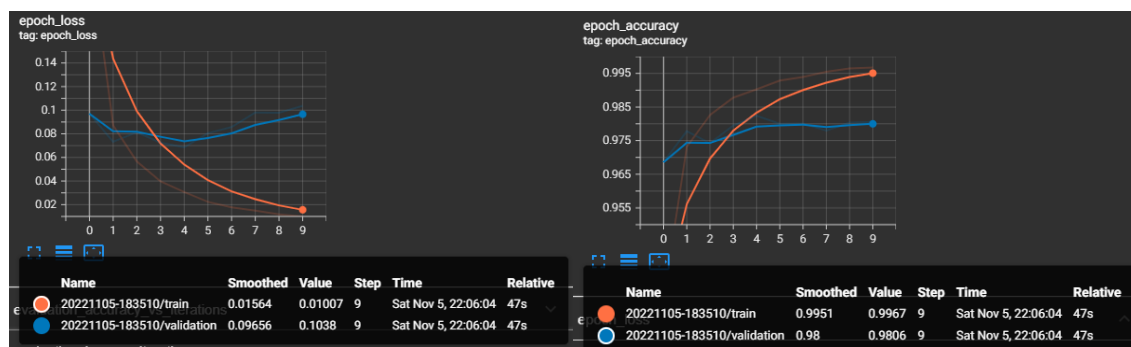
شبکه ای که تعریف می کنیم به این صورت است.

```
def build_model():  
    model = Sequential()  
    model.add(layers.Flatten(input_shape=(IMG_WIDTH, IMG_HEIGHT)))  
    model.add(layers.Dense(512, activation='relu'))  
    model.add(layers.Dense(128, activation='relu'))  
    model.add(layers.Dense(10, activation='softmax'))  
    return model
```

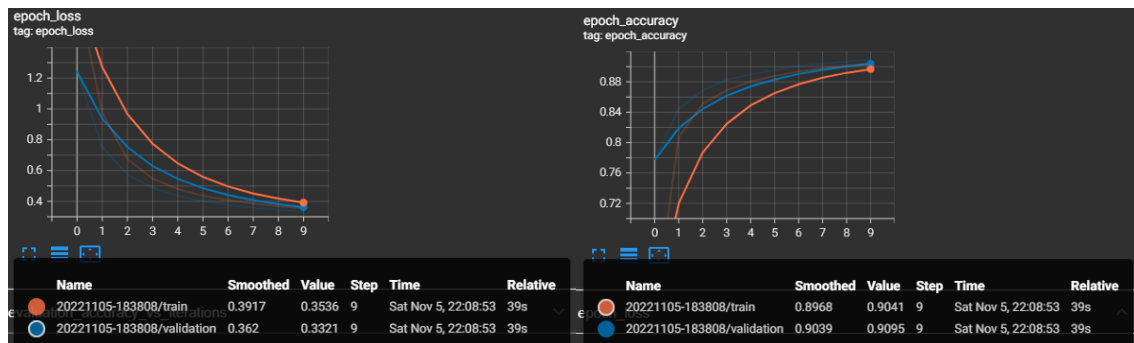
برای adam نمودار tensorboard به این گونه است



:RMSProp



:AdaGrad



با توجه به مقادیر بدست آمده RMSProp بهتر عمل کرده است پس این مدل را سیو می کنیم.

۴-

برای این مسئله چون فقط ۲ کلاس داریم پس مسئله binary classification است. پس برای لایه آخر از یک نورون استفاده می کنیم و اکتیویشن فانکشن مسئله binary classification در لایه آخر نیز تابع sigmoid است، همچنین تابع ضرر نیز binary_crossentropy است. دقت و loss مدل به این صورت است:

