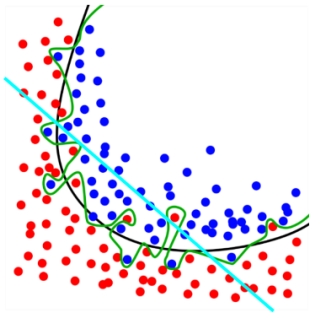
به نام خدا

تمرین هفتم یادگیری عمیق

غزل زمانی‌نژاد

۹۷۵۲۲۱۶۶

1. **اما مدلی که underfit شده است مشکل bias دارد. یعنی شبکه هنوز تمام الگوهای مرتبط با مسئله مورد نظر در داده‌های آموزشی را یاد نگرفته است و برای طبقه بندی داده ها، یک خط ساده را یاد گرفته است. پس در اینجا bias زیاد است. در شکل زیر، خط آبی رنگ نشان دهنده مدلی است که دچار underfit شده.**
مدلی که overfit شده در واقع با مشکل variance رو به رو است. بدین معنی که مدل یک منحنی پیچیده را برای طبقه بندی یاد گرفته و دارای واریانس زیاد است. در شکل زیر، منحنی سبز رنگ نشان دهنده مدلی است که دچار underfit شده.



رفع مشکل bias:

- **میتوانیم برای آموزش از یک مدل بزرگتر که دارای پارامترهای بیشتر است استفاده کنیم. وقتی تعداد پارامترها زیادتر باشد ظرفیت شبکه برای یادگیری بیشتر می شود.**
- **میتوانیم شبکه را برای مدت زمان بیشتری (یعنی تعداد بیشتری epoch) آموزش دهیم. ممکن است در مواردی (مثل استفاده از تابع خطای MSE) شبکه آهسته تر آموزش ببیند که در این موارد با تعداد کم epoch، مدل underfit میشود.**

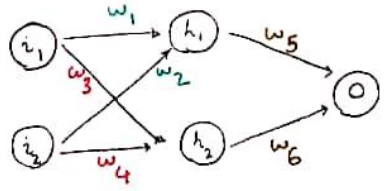
رفع مشکل variance:

- **میتوانیم ابعاد شبکه را کاهش دهیم. شبکه ای که دارای تعداد زیادی پارامتر است، ظرفیت یادگیری زیادی دارد و در نتیجه به حفظ داده ها می پردازد و تعمیم دهی خوبی ندارد.**
- **استفاده از روش های** regularization **مانند:**
  - **استفاده از** data augmentation **برای تولید داده های بیشتر برای آموزش: برای این کار روش های متعددی وجود دارد از جمله: افزودن نویز به داده،** flip**،** rotate **و ...**
  - **منظم سازی پارامتر** l1 **و** l2**: این دو روش برای جریمه اندازه پارامترها به کار می روند.**
  - Dropout**: در هر مرحله از آموزش، با احتمال** p **و به صورت تصادفی تعدادی از نورون ها را** 0 **کنیم. با این کار شبکه مجبور می شود از تمامی ویژگی های داده برای آموزش استفاده کند و تنها به برخی از ویژگی ها تکیه نکند.**

باید دقت کنیم که بین مشکل bias **و** variance **همواره یک** tradeoff **وجود دارد. مثلا در صورتی که ابعاد شبکه را بسیار زیاد کنیم واریانس زیاد می شود و در صورتی که شبکه را بسیار کوچک کنیم، بایاس زیاد میشود. پس باید بین آن تعادل برقرار کنیم.**

2. .

2)



random initialization:

$w_1 = 1 \quad w_2 = 2$

$w_3 = 1.5 \quad w_4 = -1$

$w_5 = -0.5 \quad w_6 = +1$

i_1	i_2	y_n
3	2	8
15	12	20

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Adam: $v_{dw} = \beta_1 v_{dw} + (1-\beta_1)\,dw \rightarrow v^{corrected}_{dw} = \frac{v_{dw}}{(1-\beta_1^t)}$ (momentum)

$S_{dw} = \beta_2 S_{dw} + (1-\beta_2)(dw)^2 \rightarrow S^{Corrected}_{dw} = \frac{S_{dw}}{(1-\beta_2^t)}$ (RMSprop)

$$w = w - \alpha \frac{v^{corrected}_{dw}}{\sqrt{S^{corrected}_{dw}} + \varepsilon}$$

$\beta_1 = 0.9$
$\beta_2 = 0.99$
$\alpha = 10^{-3}$
$\varepsilon = 10^{-7}$

L2 regularization: $\tilde{J} = \frac{\lambda}{2} w^T w + J$

$\partial\tilde{J}/\partial w = \lambda w + \partial J/\partial w$

$w = (1-\alpha\lambda)w - \alpha\, \partial J/\partial w$

$\lambda = 1/5 \rightarrow$ (بهتر است ضریب L2 مقدار نگاریتمی بین 0 و 0.1 باشد. بنابراین جایی راحت‌تر است محاسبات $\lambda = 1/5$)

Relu تابع $f'(x) = \begin{cases} 1 & x>0 \\ 0 & x\le 0 \end{cases}$ استفاده می‌شود. ReLU با توجه به مقادیر y_n، در لایه آخر هم به تابع فعال‌ساز

epoch 1

data 1: $z_1 = w_1 i_1 + w_2 i_2 = 1(3) + 2(2) = 7 \quad z_2 = w_3 i_1 + w_4 i_2 = \underbrace{1.5(3)}_{4.5} + 2(2) = 8.5$

$h_1 = Relu(z_1) = 7 \quad h_2 = Relu(z_2) = 8.5$

$z_o = h_1 w_5 + h_2 w_6 = \underbrace{-0.5(7)}_{-3.5} + 1(8.5) = 5$

$o = Relu(z_o) = 5$

$loss = (5-8)^2 = 9$

data 2: $z_1 = 1(15) + 2(12) = 39 \quad z_2 = \overbrace{1.5(15)}^{22.5} + (-1)(12) = 10.5$

$h_1 = Relu(z_1) = 39 \quad h_2 = Relu(z_2) = 10.5$

$z_o = 39(-0.5) + 10.5(1) = -9 \quad o = Relu(z_o) = 0$

$loss = (20-0)^2 = 400$

$J = \frac{1}{2}(9+400) = 204.5$ $\quad \tilde{J} = \frac{1}{10}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix}[w_1\ w_2\ w_3\ w_4\ w_5\ w_6] + 204.5 = 205.45$

$\frac{\partial \tilde{J}}{\partial w_1} = \lambda w_1 + \frac{\partial J}{\partial w_1}$

$\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial o}\frac{\partial o}{\partial z_0}\frac{\partial z_0}{\partial h_1}\frac{\partial h_1}{\partial z_1}\frac{\partial z_1}{\partial w_1} = \frac{-2}{2}(y-a)\cdot f'(x)\cdot w_5 \cdot f'(x)\cdot i_1$ (ReLU derivative)

$= -\left[\underbrace{(8-5)}_{3} \times 1 \times \underbrace{(-0.5)}_{-1.5} \times 3 + \underbrace{(20-0)}_{} \times 0 \times \underbrace{(-0.5)}_{-7.5} \times 15\right] = +4.5$

$\frac{\partial \tilde{J}}{\partial w_1} = \frac{1}{5}(1) + 4.5 = \boxed{4.7}$

$\frac{\partial \tilde{J}}{\partial w_2} = \lambda w_2 + \frac{\partial J}{\partial w_2}$

$\frac{\partial J}{\partial w_2} = \frac{\partial J}{\partial o}\frac{\partial o}{\partial z_0}\frac{\partial z_0}{\partial h_1}\frac{\partial h_1}{\partial z_1}\frac{\partial z_1}{\partial w_2} = \frac{-2}{2}(y-o)\, f'(x)\, w_5 \cdot f'(x)\, i_2$

$= -\left[\underbrace{(8-5)}_{3} \times 1 \times \underbrace{(-0.5)}_{-1} 2 + (20-0) \times 0 \times \underbrace{(-0.5)}_{-6} \times 12\right] = 3$

$\frac{\partial \tilde{J}}{\partial w_2} = \frac{1}{5}(2) + 3 = \boxed{3.4}$

$\frac{\partial \tilde{J}}{\partial w_3} = \lambda w_3 + \frac{\partial J}{\partial w_3}$

$\frac{\partial J}{\partial w_3} = \frac{\partial J}{\partial o}\frac{\partial o}{\partial z_0}\frac{\partial z_0}{\partial h_2}\frac{\partial h_2}{\partial z_2}\frac{\partial z_2}{\partial w_3} = \frac{-2}{2}(y-a)\cdot f'(x)\, w_6\, f'(x)\, i_1$

$= -\left[\underbrace{(8-5)}_{3} \times 1 \times 1 \times 3 + (20-0) \times 0 \times 1 \times 15\right] = -9$

$\frac{\partial \tilde{J}}{\partial w_3} = \underbrace{\frac{1}{5}}_{0.3}(1.5) + (-9) = \boxed{-8.7}$

$$\frac{\partial \tilde{J}}{\partial w_4} = \lambda w_4 + \frac{\partial J}{\partial w_4}$$

$$\frac{\partial J}{\partial w_4} = \frac{\partial J}{\partial o} \frac{\partial o}{\partial z_o} \frac{\partial z_o}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial w_4} = \frac{-2}{2}(y-a) f'(x) w_6 f'(x) i_2$$

$$= -\left[\underbrace{(8-5)}_{3} \times 1 \times 1 \times 2 + (20.0) \times 0 \times 1 \times 12 \right] = -6$$

$$\frac{\partial \tilde{J}}{\partial w_4} = \frac{1}{5}(-1) + (-6) = \boxed{-6.2}$$

$$\frac{\partial \tilde{J}}{\partial w_5} = \lambda w_5 + \frac{\partial J}{\partial w_5}$$

$$\frac{\partial J}{\partial w_5} = \frac{\partial J}{\partial o} \frac{\partial o}{\partial z_o} \frac{\partial z_o}{\partial w_5} = \frac{-2}{2}(y-a) f'(x) h_1$$

$$= -\left[\underbrace{(8-5)}_{3} \times 1 \times 7 + (20-0) \times 0 \times 39 \right] = -21$$

$$\frac{\partial \tilde{J}}{\partial w_5} = \frac{1}{5}(-0.5) + (-21) = \boxed{-21.1}$$

$$\frac{\partial \tilde{J}}{\partial w_6} = \lambda w_6 + \frac{\partial J}{\partial w_6}$$

$$\frac{\partial J}{\partial w_6} = \frac{\partial J}{\partial o} \frac{\partial o}{\partial z_o} \frac{\partial z_o}{\partial w_6} = \frac{-2}{2}(y-a) f'(x) h_2 = -\left[\underbrace{(8-5)}_{3} \times 1 \times 8.5 + (20-0) \times 0 \times 10.5 \right] = -25.5$$

$$\frac{\partial \tilde{J}}{\partial w_6} = \frac{1}{5}(1) + (-25.5) = \boxed{-25.3}$$

update w_1: $v = 0.9 \times 0 + 0.1(4.7) = 0.47$ $\quad v_{corrected} = \frac{0.47}{1-0.9^1} = 4.7$

$s = 0.99 \times 0 + 0.01(4.7)^2 = 0.2209$ $\quad s_{Corrected} = \frac{0.2209}{1-0.99} = 22.09$

$$w_1 = 1 - 10^{-3} \frac{4.7}{\sqrt{22.09 + 10^{-7}}} = \boxed{0.999}$$

update w_2: $v = 0.9 \times 0 + 0.1(3.4) = 0.34$ $\quad v_{corr} = \frac{0.34}{1-0.9^1} = 3.4$

$S = 0.99 \times 0 + 0.01(3.4)^2 = 0.1156$ $\quad S_{corr} = \frac{0.1156}{1-0.99^1} = 11.56$

$$w_2 = 2 - 10^{-3} \frac{3.4}{\sqrt{11.56 + 10^{-7}}} = \boxed{1.999}$$

update w_3: $v = 0.9 \times 0 + 0.1(-8.7) = -0.87$ $\quad v_{corr} = \frac{-0.87}{1-0.9^1} = -8.7$

$S = 0.99 \times 0 + 0.01(-8.7)^2 = 0.7569$ $\quad S_{corr} = \frac{0.7569}{1-0.99} = 75.69$

$$w_3 = 1.5 - 10^{-3} \frac{-8.7}{\sqrt{75.69 + 10^{-7}}} = \boxed{1.501}$$

update w_4: $v = 0.9 \times 0 + 0.1(-6.2) = -0.62$ $\quad v_{corr} = \frac{-0.62}{1-0.9} = -6.2$

$S = 0.99 \times 0 + 0.01(-6.2)^2 = 0.3844$ $\quad S_{corr} = \frac{0.3844}{1-0.99^1} = 38.44$

$$w_4 = -1 - 10^{-3} \frac{-6.2}{\sqrt{38.44 + 10^{-7}}} = \boxed{-0.999}$$

update w_5: $v = 0.9 \times 0 + 0.1(-21.1) = -2.11$ $\quad v_{corr} = \frac{-2.11}{1-0.9^1} = -21.1$

$S = 0.99 \times 0 + 0.01(-21.1)^2 = 4.4521$ $\quad S_{corr} = \frac{4.4521}{1-0.99^1} = 445.21$

$$w_5 = -0.5 - 10^{-3} \frac{-21.1}{\sqrt{445.21 + 10^{-7}}} = \boxed{-0.499}$$

update w_6: $v = 0.9 \times 0 + 0.1(-25.3) = -2.53$ $\quad v_{corr} = \frac{-2.53}{1-0.9^1} = -25.3$

$S = 0.99 \times 0 + 0.01(-25.3)^2 = 6.4009$ $\quad S_{corr} = \frac{6.4009}{1-0.99^1} = 640.09$

$$w_6 = +1 - 10^{-3} \frac{-25.3}{\sqrt{640.09 + 10^{-7}}} = \boxed{1.001}$$

updated weights: $w_1 = 0.999$, $w_2 = 1.999$, $w_3 = 1.501$, $w_4 = -0.999$

, $w_5 = -0.499$, $w_6 = 1.001$

epoch 2 در این epoch از وزن‌هایی که در epoch قبل استاد شد مجددا استفاده می‌کنیم.

data 1: $z_1 = 0.999(3) + 1.999(2) = 6.995$ $\quad z_2 = 1.501(3) + (-0.999)\times 2 = 2.505$

$h_1 = 6.995$ $\quad h_2 = 2.505$

$z_o = 6.995(-0.499) + 2.505(1.001) = -0.983 \rightarrow o = 0$

$loss = (8-0)^2 = 64$

data 2: $z_1 = 0.999(15) + 1.999(12) = 38.973$ $\quad z_2 = 1.501(15) + (-0.999)(12) = 10.527$

$h_1 = 38.973$ $\quad h_2 = 10.527$

$z_o = 38.973(-0.499) + 10.527(1.001) = -8.91 \rightarrow o = 0$

$loss = (20-0)^2 = 400$

$J = \frac{1}{2}[64 + 400] = 232$ $\quad \tilde{J} = \frac{1}{10}|w|_2^2 + 232 \simeq 241.49$

با توجه به اینکه در تمامی گرادیان‌های J نسبت به وزن‌ها یک ترم Relu derivative داریم و چون داخل پرانتز هم خروجی داده اول و هم خروجی داده دوم صفر شد، پس:

$(Relu(x))' = 0 \Rightarrow \frac{\partial J}{\partial w} = 0$

$\Rightarrow \frac{\partial \tilde{J}}{\partial w} = \lambda w + 0 = \lambda w$

$\frac{\partial \tilde{J}}{\partial w_1} = \frac{1}{5}\,0.999 = 0.1998$ $\quad \frac{\partial \tilde{J}}{\partial w_4} = \frac{1}{5}(-0.999) = -0.1998$

$\frac{\partial \tilde{J}}{\partial w_2} = \frac{1}{5}(1.999) = 0.3998$ $\quad \frac{\partial \tilde{J}}{\partial w_5} = \frac{1}{5}(-0.499) = -0.0998$

$\frac{\partial \tilde{J}}{\partial w_3} = \frac{1}{5}(1.501) = 0.3002$ $\quad \frac{\partial \tilde{J}}{\partial w_6} = \frac{1}{5}(1.001) = 0.2002$

update w_1: $v = 0.9 \times 0.47 + 0.1(0.1998) \simeq 0.442$ $\quad v_{corr} = \frac{0.442}{1-0.9^2} \simeq 2.32$

$S = 0.99 \times 0.22 + 0.01(0.1998)^2 \simeq 0.218$ $\quad S_{corr} = \frac{0.218}{1-0.99^2} = 10.95$

$$w_1 = 0.999 - 10^{-3}\,\frac{2.36}{\sqrt{10.95 + 10^{-7}}} \simeq 0.998$$

update w_2: $v = 0.9(0.34) + 0.1(0.3998) \simeq 0.345$

$S = 0.99(0.1156) + 0.01(0.3998)^2 \simeq 0.116$

$v_{Corr} = \frac{0.345}{1-0.9^2} = 1.8211$

$S_{Corr} = \frac{0.116}{1-0.99^2} \simeq 5.829$

$w_2 = 1.999 - 10^{-3} \frac{v_{Corr}}{\sqrt{S_{Corr}} + 10^{-7}} \simeq \boxed{1.998}$

update w_3: $v = 0.9(-0.87) + 0.1(0.3002) \simeq 0.753$

$S = 0.99(0.7569) + 0.01(0.3002)^2 \simeq 0.750$

$v_{Corr} = \frac{0.753}{1-0.9^2} \simeq 3.963$

$S_{Corr} = \frac{0.750}{1-0.99^2} \simeq 37.698$

$w_3 = 1.501 - 10^{-3} \frac{v_{Corr}}{\sqrt{S_{Corr}} + 10^{-7}} \simeq \boxed{1.5016}$

update w_4: $v = 0.9(-0.62) + 0.1(-0.1998) \simeq -0.577$

$S = 0.99(0.3844) + 0.01(-0.1998)^2 \simeq 0.380$

$v_{Corr} = \frac{-0.577}{1-0.9^2} \simeq -3.036$

$S_{Corr} = \frac{0.38}{1-0.99^2} \simeq 19.095$

$w_4 = -0.999 - 10^{-3} \frac{v_{Corr}}{\sqrt{S_{Corr}} + 10^{-7}} \simeq \boxed{-0.998}$

update w_5: $v = 0.9(-2.11) + 0.1(-0.0998) \simeq -1.908$

$S = 0.99(4.4521) + 0.01(-0.0998)^2 \simeq 4.407$

$v_{Corr} = \frac{-1.908}{1-0.9^2} \simeq -10.042$

$S_{Corr} = \frac{4.407}{1-0.99^2} \simeq 221.457$

$w_5 = -0.499 - 10^{-3} \frac{v_{Corr}}{\sqrt{S_{Corr}} + 10^{-7}} \simeq \boxed{-0.498}$

update w_6: $v = 0.9(-2.53) + 0.1(0.2002) \simeq -2.256$

$S = 0.99(6.4009) + 0.01(0.2002)^2 \simeq 6.337$

$v_{Corr} = \frac{-2.256}{1-0.9^2} \simeq -11.873$

$S_{Corr} = \frac{6.337}{1-0.99^2} \simeq 318.44$

$w_6 = 1.001 - 10^{-3} \frac{v_{Corr}}{\sqrt{S_{Corr}} + 10^{-7}} \simeq \boxed{1.0016}$

برای آموزش مدل تنها از 2 epoch استفاده شد و این مقدار بسیار کم است. مقدار تابع ضرر همچنان کمتر میشود یعنی مدل هنوز به همگرایی نرسیده است. با توجه به استفاده از تابع ضرر MSE چون دیرتر به همگرایی میرسیم بهتر است آن را در epochهای بیشتری آموزش دهیم.

3. **الف) در این نوت بوک، مدلی برای دیتاست Higgs آورده شده و چند روش منظم سازی بر روی آن اعمال شده که به بررسی آنها می پردازیم:**

- **کم کردن ظرفیت شبکه: ساده ترین راه جلوگیری از regularization کم کردن تعداد پارامترهای شبکه است. در آن صورت شبکه کمتر الگوها را حفظ میکند و میتواند تعمیم دهی بهتری داشته باشد.**

در این قسمت 4 مدل با تعداد پارامترهای مختلف داریم:

مدل اول بسیار کوچک است و تنها شامل دو لایه است.

Train_accuracy = .67, val_accuracy = .654

مدل دوم کوچک است و شامل دو لایه مخفی و در کل 3لایه است.

Train_accuracy =.69, val_accuracy = .657

مدل سوم متوسط است و شامل سه لایه مخفی است.

Train_accuracy = .79, val_accuracy = .64

مدل چهارم بزرگ است و شامل چهار لایه مخفی است.

Train_accuracy = 1, val_accuracy = .65

در این مدل ها، مدل 4 بهترین دقت را روی داده آموزشی داشته و توانسته تمام داده های آموزشی را به درستی طبقه بندی کند. اما اختلاف میان دقت آموزش و آزمون بسیار زیاد است. یعنی با زیاد کردن تعداد پارامترها مدل overfit شده و نتوانسته به خوبی تعمیم داشته باشد. به همین دلیل دقت داده آزمون بالا نرفته است.

مدل دوم بهترین دقت را روی داده آزمون کسب کرده و اختلاف میان دقت آموزش و آزمون زیاد نیست. اما دقت آموزش خیلی زیاد نیست و میتوان گفت این شبکه دچار underfit شده است.

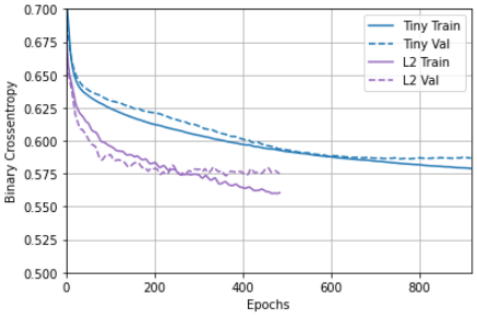
در کل مطابق انتظار کم کردن ظرفیت شبکه منجر به بالاتر رفتن دقت آزمون شده است (اگر ظرفیت شبکه خیلی کم (tiny model) باشد نتیجه بهتری نخواهیم گرفت پس بهتر است بین ظرفیت شبکه و دقت آزمون تعادل برقرار کنیم).

- **اضافه کردن جریمه به پارامترها: یکی از روش های منظم سازی افزودن نورم l1 یا l2 به مقدار loss است. با این کار وزن های شبکه مجبور می شوند مقادیر کوچکی داشته باشند و از overfit جلوگیری می شود. در این قسمت یک مدل با L2 regularization ساخته و آموزش داده شده.**

Train_accuracy =.6934, val_accuracy = .6690

در این قسمت از همان مدل large قسمت قبل استفاده شد و l2 به آن اضافه شد. دقت داده آموزشی از 1 به نزدیک 70 درصد رسید اما دقت آزمون بالاتر رفت و اختلاف میان این دو دقت بسیار کم شد. یعنی با افزودن نورم L2 شبکه دیگر overfit نشد.

اگر نمودار تابع ضرر آن را با مدل tiny مقایسه کنیم، دیده میشود که این دو نمودار نسبت به حالت قبل بهم نزدیکتر شده اند. یعنی با اضافه کردن ترم L2 به ضرر، با وجود بیشتر شدن مقدار loss نسبت به حالت قبل و همچنین داشتن تعداد پارامترهای مشابه، شبکه تعمیم بهتری داشته.

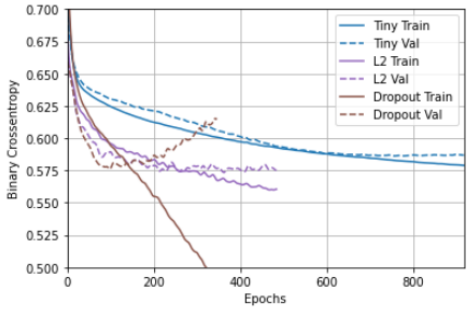


- افزودن dropout: در هر مرحله از آموزش، با احتمالی تعدادی از نورون ها را خاموش میکنیم و مقدار آن را 0 در نظر میگیریم. با این کار شبکه مجبور میشود برای آموزش از تمامی ویژگی ها استفاده کند و نتواند الگوهای خاصی را حفظ کند. در نتیجه از overfit پیشگیری میشود.

در مدل این قسمت از مدل large استفاده شده اما بعد از هر لایه Dense، از یک لایه Dropout استفاده شده. نتایج آن:

Train_accuracy = .72, val_accuracy = .68

دقت آزمون نسبت به مدل بزرگ اولیه 3 درصد بهتر شده است. یعنی اضافه کردن dropout توانسته مقداری از overfit جلوگیری کند. اما مطابق نمودار زیر، همچنان نتوانسته عملکردی به خوبی tiny model داشته باشد.

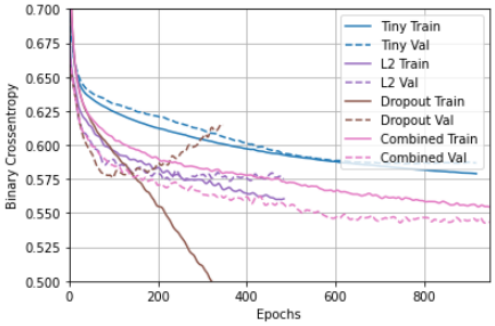


- L2 + Dropout: در این مدل بر روی مدل بزرگ هم L2 و هم dropout را پیاده سازی میکنیم. نتایج بدست آمده:

Train_accuracy = .6994, val_accuracy = .69

این مدل موفق شده بالاترین درصد دقت داده آزمون را کسب کند و تعمیم دهی آن از سایر مدل ها بهتر بوده.

نمودار هر 3 حالت Regularization **در مقایسه با مدل** tiny **در تصویر زیر مشاهده میشود:**



ب) ابتدا برای اینکه بتوانیم از امکانات تنسوربورد استفاده کنیم در بعضی از سلول ها از جمله get_callbacks **کمی تغییر ایجاد میکنیم. برای اینکه شبکه از** underfit **خارج شود ابتدا شبکه را با ابعاد مختلف امتحان میکنیم و نتیجه تمام مدل ها را در دیکشنری** sizes **ذخیره می کنیم. بررسی مدل ها:**

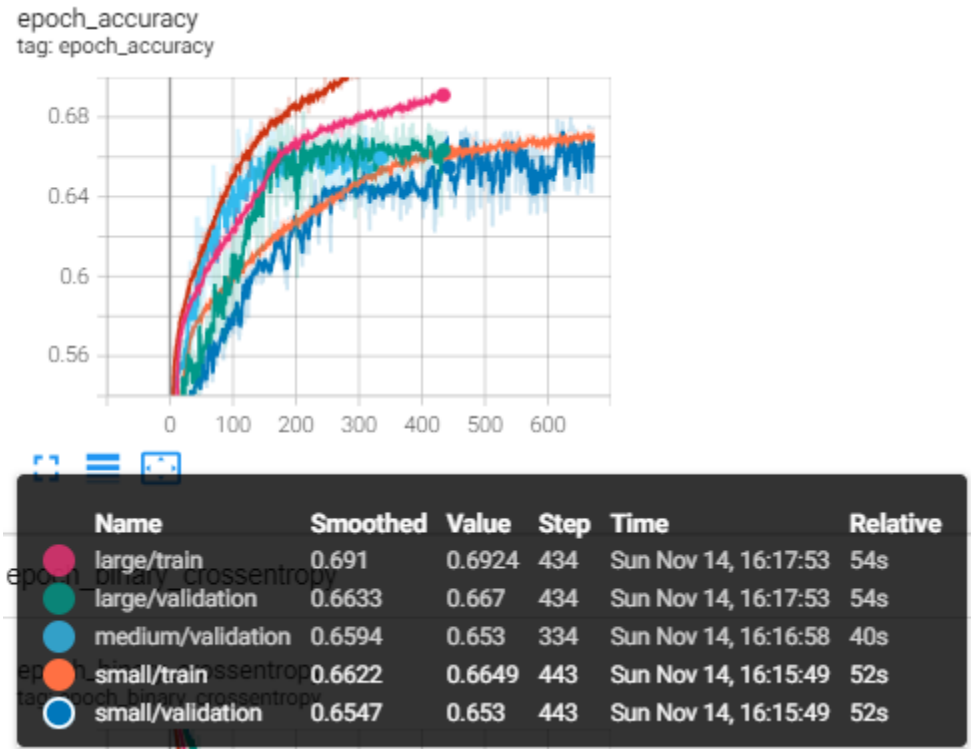
- **مدل** small **که از قبل پیاده سازی شده است.**
- **مدل** medium **با 3 لایه مخفی** dense **و تابع فعال سازی** elu **پیاده سازی شده است.**

```
model2 = tf.keras.Sequential([
    layers.Dense(16, activation='elu', input_shape=(FEATURES,)),
    layers.Dense(32, activation='elu'),
    layers.Dense(64, activation='elu'),
    layers.Dense(1)
])
```

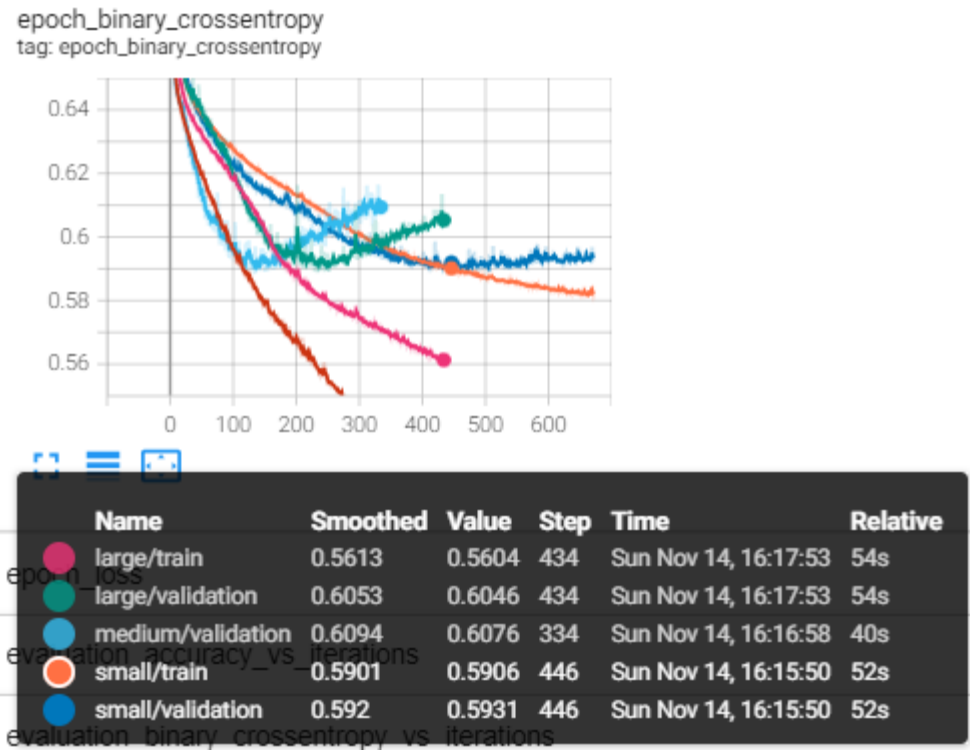
- **مدل** large **با 4 لایه مخفی** dense **و تابع فعال سازی** elu **و تابع سیگموید برای لایه آخر پیاده سازی شده است.**

```
model3 = tf.keras.Sequential([
    layers.Dense(8, activation='elu', input_shape=(FEATURES,)),
    layers.Dense(16, activation='elu'),
    layers.Dense(32, activation='elu'),
    layers.Dense(64, activation='sigmoid'),
    layers.Dense(1)
])
```

این 3 مدل را با استفاده از متد compile_and_fit **کامپایل کرده و آموزش میدهیم. نمودار دقت و خطای این 3 مدل در تنسوربورد مطابق شکل زیر است:**



در اینجا بهترین دقت داده آموزشی مربوط به مدل medium است اما دقت داده آزمون آن با دقت آموزشی اختلاف زیادی دارد. یعنی مدل medium تاحدی overfit شده. مدل large که دارای تعداد پارامتر بیشتری نسبت به مدل small است بهترین دقت آزمون را بدست آورده است. همچنان میان دقت آموزش و آزمون این مدل حدود 5 درصد اختلاف وجود دارد. پس این مدل را برای قسمت های بعدی انتخاب میکنیم و سعی میکنیم با استفاده از روش های منظم سازی، دقت آزمون بهتر شده و به آموزش نزدیکتر شود.



نمودار خطای مدل small در آموزش و آزمون از سایر مدل ها بهم نزدیکتر است. چون ظرفیت آموزش شبکه کم بوده و شبکه عملا underfit شده است. اما در مدل medium بعد از 120 epoch و در مدل large بعد از تقریبا 250 epoch مقدار ضرر بیشتر شده است. به این دلیل که بعد از آن نقطه شبکه به سمت overfit شدن پیش رفته است.

- **مدل large با استفاده از L2: بر روی همان مدل قبلی L2 regularization را اعمال میکنیم تا شبکه وادار شود وزن های کوچکتری یاد بگیرد و کمتر overfit شود.**

```
model_l2 = tf.keras.Sequential([
    layers.Dense(8, activation='elu', input_shape=(FEATURES,), kernel_regularizer=regularizers.l2(0.001)),
    layers.Dense(16, activation='elu', kernel_regularizer=regularizers.l2(0.001)),
    layers.Dense(32, activation='elu', kernel_regularizer=regularizers.l2(0.001)),
    layers.Dense(64, activation='sigmoid', kernel_regularizer=regularizers.l2(0.001)),
    layers.Dense(1)
])
```

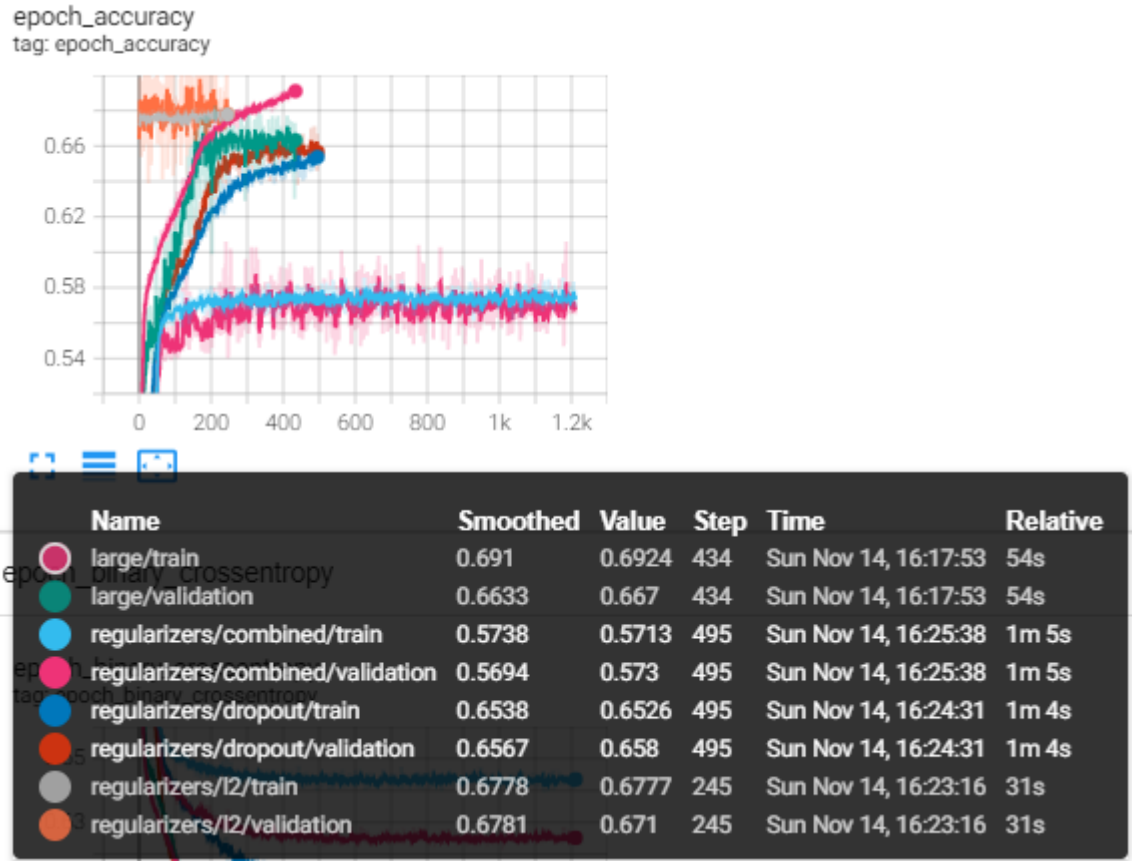
- **مدل** large **با استفاده از** Dropout**: بعد از هریک از لایه های مخفی از لایه** dropout **با احتمال** 0.3 **استفاده میکنیم. یعنی در هر مرحله آموزش،** 0.3 **از نورون های هر لایه به صورت تصادفی خاموش میشوند تا مدل کمتر به ویژگی های خاص تکیه کند و به کاهش** overfit **کمک کند.**

```
model_dropout = tf.keras.Sequential([
    layers.Dense(8, activation='elu', input_shape=(FEATURES,)),
    layers.Dropout(0.3),
    layers.Dense(16, activation='elu'),
    layers.Dropout(0.3),
    layers.Dense(32, activation='elu'),
    layers.Dropout(0.3),
    layers.Dense(64, activation='sigmoid'),
    layers.Dropout(0.3),
    layers.Dense(1)
])
```

- **مدل** large **با استفاده از** Dropout **و** L2**: از هر دو روش** regularization **استفاده میکنیم تا عملکرد** regularization 2 **را ببینیم.**

```
model_combined = tf.keras.Sequential([
    layers.Dense(8, activation='elu', input_shape=(FEATURES,), kernel_regularizer=regularizers.l2(0.001)),
    layers.Dropout(0.3),
    layers.Dense(16, activation='elu', kernel_regularizer=regularizers.l2(0.001)),
    layers.Dropout(0.3),
    layers.Dense(32, activation='elu', kernel_regularizer=regularizers.l2(0.001)),
    layers.Dropout(0.3),
    layers.Dense(64, activation='sigmoid', kernel_regularizer=regularizers.l2(0.001)),
    layers.Dropout(0.3),
    layers.Dense(1)
])
```

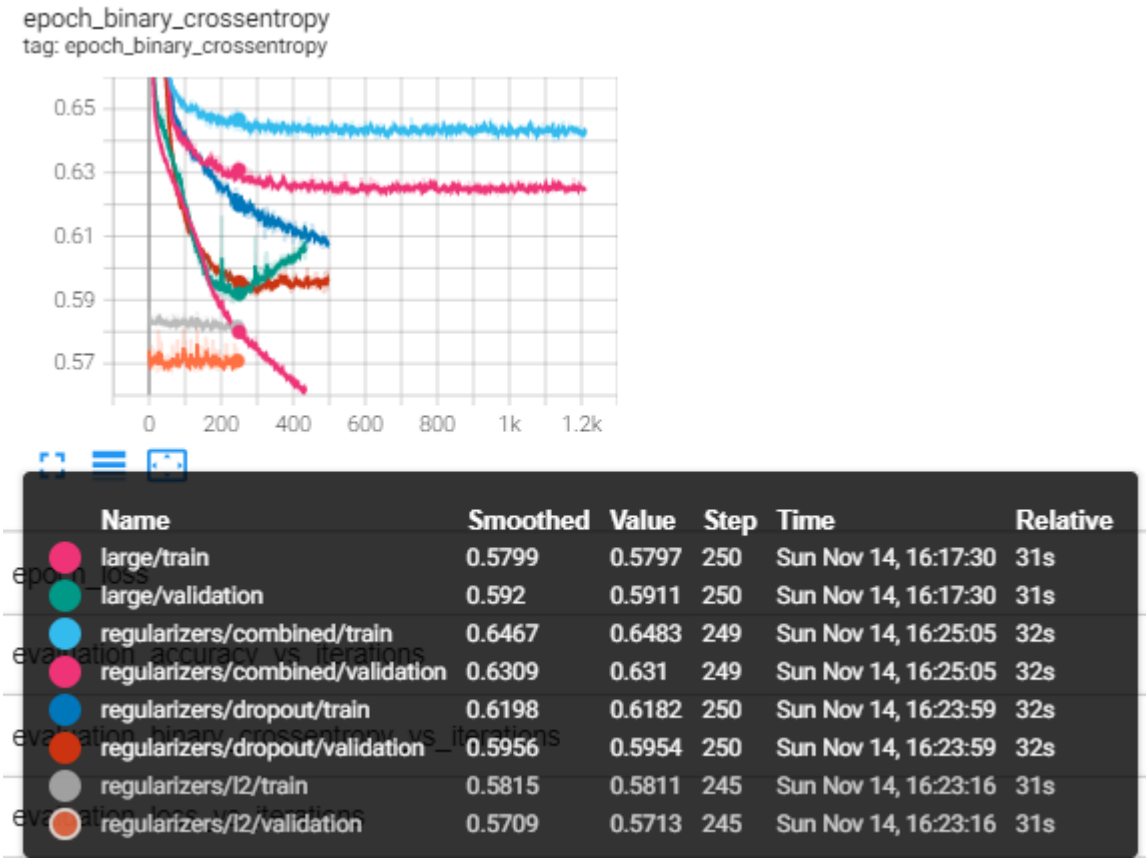
مدل ها را با استفاده از متد compile_and_fit **کامپایل کرده و آموزش میدهیم. نمودار دقت و خطای این مدل ها در تنسوربورد مطابق شکل زیر است:**



در این بخش بهترین عملکرد دقت داده آزمون مربوط به نمودار نارنجی رنگ یعنی استفاده از L2 regularization است. نمودار نارنجی و خاکستری که به یکدیگر مرتبط هستند تقریبا عملکرد مشابه داشتند یعنی در این مدل overfit رخ نداده است.

در نمودار combined هم نمودار آموزش و آزمون به یکدیگر نزدیکند ولی مقدار دقت بسیار پایین آمده و به حالت تصادفی نزدیک شده است. این مدل به نوعی underfit شده است.

اختلاف بین نمودار آموزش و آزمون dropout نیز بسیار کم شده و این مدل هم از overfit درآمده است.



نمودار آموزش و آزمون L2 بهتر از سایر مدل ها توانسته اند میزان loss را کمینه کنند. نمودار آموزش شبکه اصلی (large) همچنان نزولی است و مدل میتواند میزان ضرر آموزش را کمتر کند. اما نمودار سبز رنگ که نمودار آزمون همین مدل است بعد از تقریبا epoch250 صعودی شده یعنی مدل اصلی تعمیم مناسبی ندارد.

شیب کاهش سرعت نمودار آموزش و آزمون در 2 مدل دیگر تقریبا به یک صورت است.

نتیجه نهایی: در این دیتاست بهتر است برای آموزش آن از یک مدل large (مثلا 4 لایه مخفی) استفاده کنیم تا underfit نشود. برای اینکه overfit نشود بهترین کار استفاده از L2 regularization است.