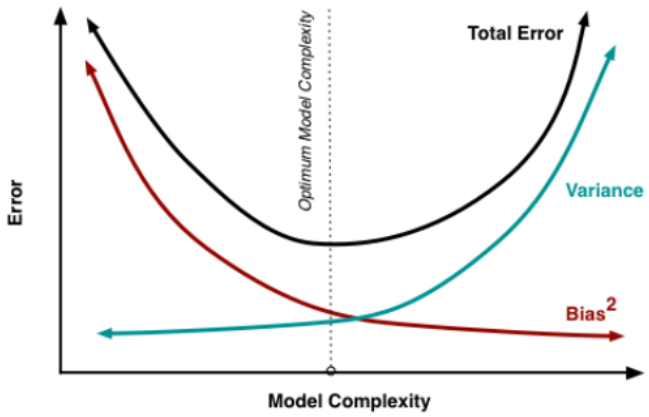
تمرین هفتم یادگیری عمیق

محمدعلی فراهت ۹۷۵۲۱۴۲۳

سوال ۱)

مدلی که overfit شده، احتمالا دارای bias پایین و variance بالا ، روی داده آموزش می‌باشد و مدلی با پیچیدگی بالاتر است. و مدلی که underfit هست، احتمالا دارای bias بالا است و بخاطر پیچیدگی پایین ، variance آن مدل کم خواهد بود. مدل بهینه برای ما باید نه overfit باشد ، و نه underfit . در شکل زیر نمودار پیچیدگی را برای bias و variance می‌بینید:



برای پیشگیری از این اتفاقات راه هایی وجود دارد. برای جلوگیری از overfitting میتوانیم از جریمه اندازه پارامترها (محدود کردن ظرفیت یادگیری)، یا منظم سازی پارامتر L2 و یا Augmentation استفاده کنیم. و برای underfitting هم باید یادگیری را با تعداد داده بیشتر و تعداد دفعات بیشتر انجام دهیم، و یا مدل را پیچیده تر کنیم (مثلا لایه بیشتری اضافه کنیم)

سوال ۲)

2) وزن‌ها را به صورت زیر انتخاب می‌کنیم:

$w_1 = w_2 = w_3 = w_4 = w_5 = w_6 = 1$

$MSE = \frac{1}{n}\sum (y-\hat{y})^2 + \frac{1}{2n}\sum w^2$ ← منظم‌سازی

iteration 1 :

forward : $h_1 = i_1 w_1 + i_2 w_2 = 5$ ← بعد از relu تغییری نمی‌کند

① $h_2 = i_1 w_3 + i_2 w_4 = 5$, $O_{in} = h_1 w_5 + h_2 w_6 = 10$

$sigmoid(O) = O_{out_1} = 0.999954 \simeq 1$

② $h_1 = 2.7$, $h_2 = 2.7$, $O_{in} = 5.4$ → $O_{out_2} = 1$

$\Rightarrow L = \frac{1}{2}((1+3)+(0+3)) = 3.5$

Back Propagation :

$w_t' = w_t - \alpha \frac{V_t}{\sqrt{S_t+\epsilon}} \times g_t$, $V_t = \beta_1 V_{t-1} - (1-\beta_1) g_t$

$S_0 = 0$, $V_0 = 0$, $S_t = \beta_2 S_{t-1} - (1-\beta_2) g_t$

گرادیان w_6 : $g_1 = \frac{\partial L}{\partial w_6} = \frac{\partial L}{\partial O_{out}} \cdot \frac{\partial O_{out}}{\partial O_{in}} \cdot \frac{\partial O_{in}}{\partial w_6}$

① $= (2 \times \frac{1}{2}((1-0)+0)(-1) = -1$

② $O_{out}(1-O_{out}) = 0 \rightarrow \boxed{g_1 = 0}$ ← چون Sigmoid استفاده کرده بودیم و در اعداد بزرگ تقریباً ۱ می‌شد یا تقریب ۵ رقم اعشار ۰ می‌شود

w_6 تغییری نمی‌کند با این مقادیر اشتباه شد

↳ w_5 همینطور ← به همین خاطر تغییری نمی‌کند

در iteration بعدی هم تغییری نمی‌کند ←

سوال ۳)

الف) در این بخش مدلی داریم که overfit شده است، در قسمت اول می‌بینیم که با افزایش پیچیدگی مدل (افزودن تعداد لایه‌ها) باعث افزایش overfitting شدیم.

اولین روش جلوگیری از overfitting که استفاده شده، روش L2 regularization است. با این کار مدلی به پیچیدگی ۴ لایه پنهان ۵۱۲ نورونی بدون overfit شدن به آموزش ادامه می‌دهد. البته تا ۴۰۰ دور آموزش این روند ادامه دارد.

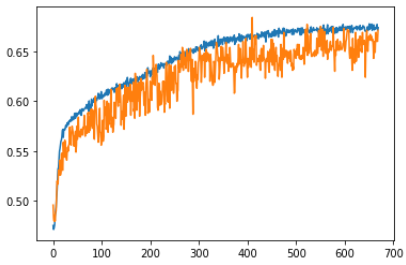
روش بعدی ، Dropout است. با این کار تا حدی جلوی overfitting گرفته شد ولی بعد از ۱۸۰ دور آموزش، دوباره مدل overfit شد. پس به تنهایی این کار کمک زیادی نمی‌کند و اثر آن فقط در اول آموزش است.

در حالت بعدی دو روش قبل باهم ترکیب شده و همزمان اجرا می‌شوند. با این کار میتوانیم با یک مدل پیچیده، بدون overfitting دیتاست خود را آموزش بدهیم و حتی از مدل های کوچک تر هم بهتر کار کند و loss کمتری داشته باشد.

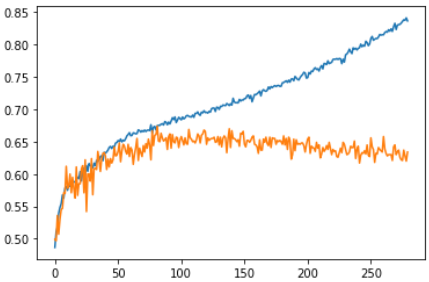
همچنین با افزایش دیتای آموزش میتوان نتیجه بهتری گرفت.

ب) برای این بخش، من ابتدا تعدادی لایه اضافه کردم تا مدل کمی پیچیده تر شود و از underfit بودن خارج شود. نمودار آن را در زیر می‌بینید:

دقت مدل کوچک (آبی آموزش و نارنجی validation است)، می‌بینیم که بسیار کم است:

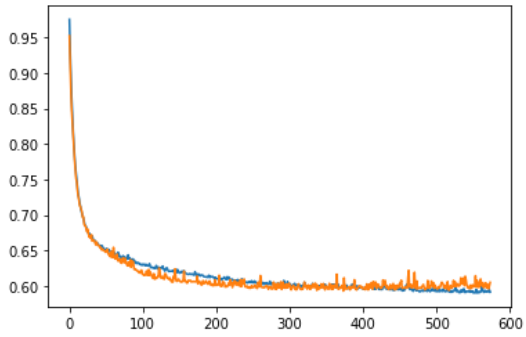


دقت مدل بزرگ و پیچیده، می‌بینیم که overfitting اتفاق افتاده:

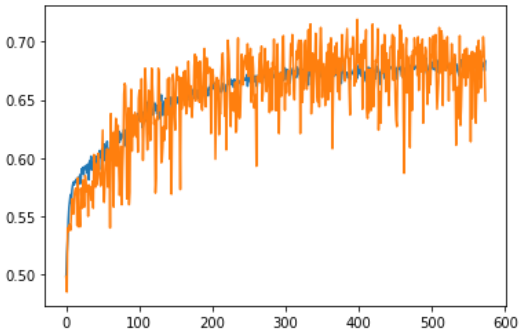


حالا L2 regularization را پیاده سازی و اجرا می‌کنیم. نتیجه به شکل زیر است :

نمودار ضرر:

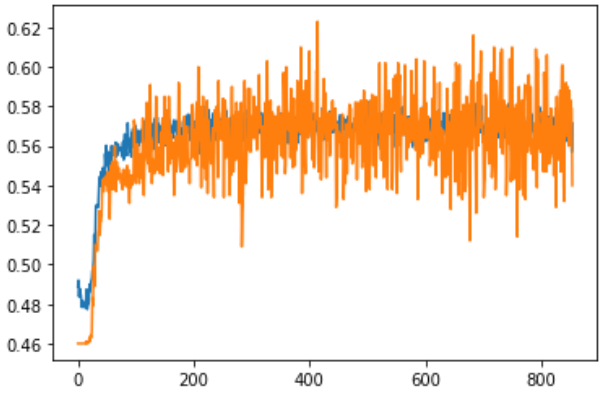


نمودار دقت:

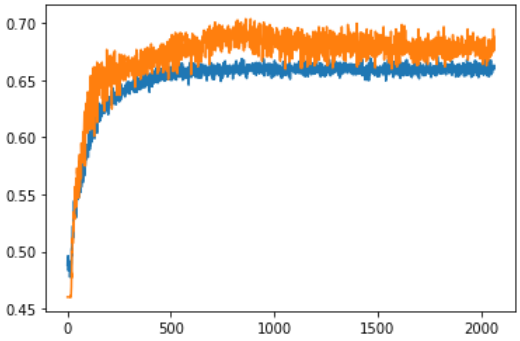


می‌بینیم که با این کار جلوی overfitting تا جای خوبی گرفته شد ولی هنوز مدل دقت خوبی ندارد.

با استفاده کردن از dropout ، نتیجه بدتر از قبل شد. نمودار دقت به صورت زیر است:



حالا dropout را همراه با regularization استفاده **نکردم**، در نمودار دقت می‌بینیم که بهبود حاصل شد.



بهترین و مناسب ترین جواب برای این سوال همان استفاده از L2 reularization به تنهایی بود که دقت آن هم برای تست و هم آموزش برابر بود و حدود ۷۰ درصد بوده.