

به نام خدا

بیش برآزش:

مدل ما توانایی تولید نتایج خوب روی داده های آموزشی را دارد ولی روی داده هایی که تا به حال ندیده است عملکرد ضعیفی دارد (داده های آموزشی را حفظ میکند). در حالیکه هدف یادگیری عمیق یا به طور کلی هدف یادگیری ماشین تعمیم دهی از روی داده هایی آموزشی به هر داده ای از دامنه مسئله میباشد. یعنی ما میخواهیم که پیش بینی های خوب روی داده هایی که تا به حال ندیده است انجام بدهد.

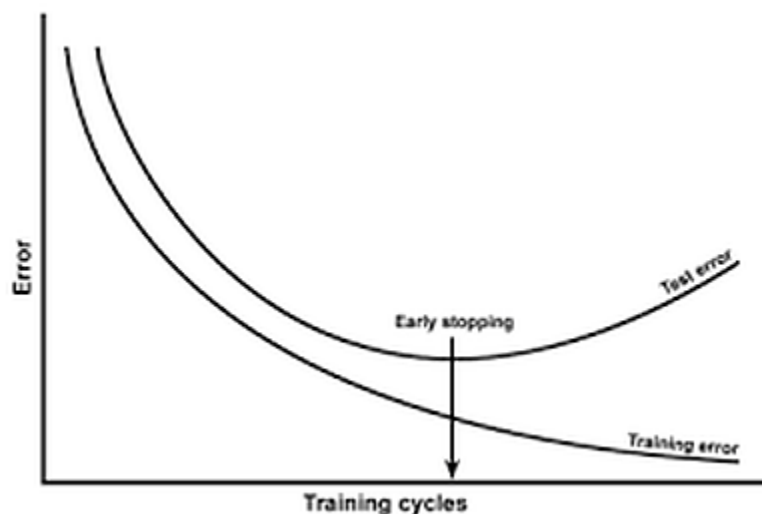
5 تکنیک برای کاهش این مشکل:

1- ساده کردن مدل:

اولین قدمی که برای کاهش بیش برآزش میتوان برداشت کاهش پیچیدگی مدل است. برای کاهش پیچیدگی مدل به سادگی میتوان 1- لایه های شبکه را حذف کرد. 2- تعداد نرون ها را کاهش داد. به طور کلی قانونی وجود ندارد که بگوید چه تعداد از لایه ها را باید حذف کرد یا شبکه ما باید چقدر بزرگ باشد. اما اگر مشاهده کردیم که شبکه ما دچار مشکل بیش برآزش شده است باید سعی کنیم آن را کوچکتر کنیم (البته تا جایی که دچار underfit نشویم)

2- Early stopping:

این راه حل یک نوع از regularization میباشد برای مدل هایی که آموزش خود را به روش های iterative انجام میدهند. چون شبکه های عصبی یادگیری را منحصر به وسیله gradient descent انجام میدهند پس این تکنیک برای همه مسئله ها کاربرد دارد.

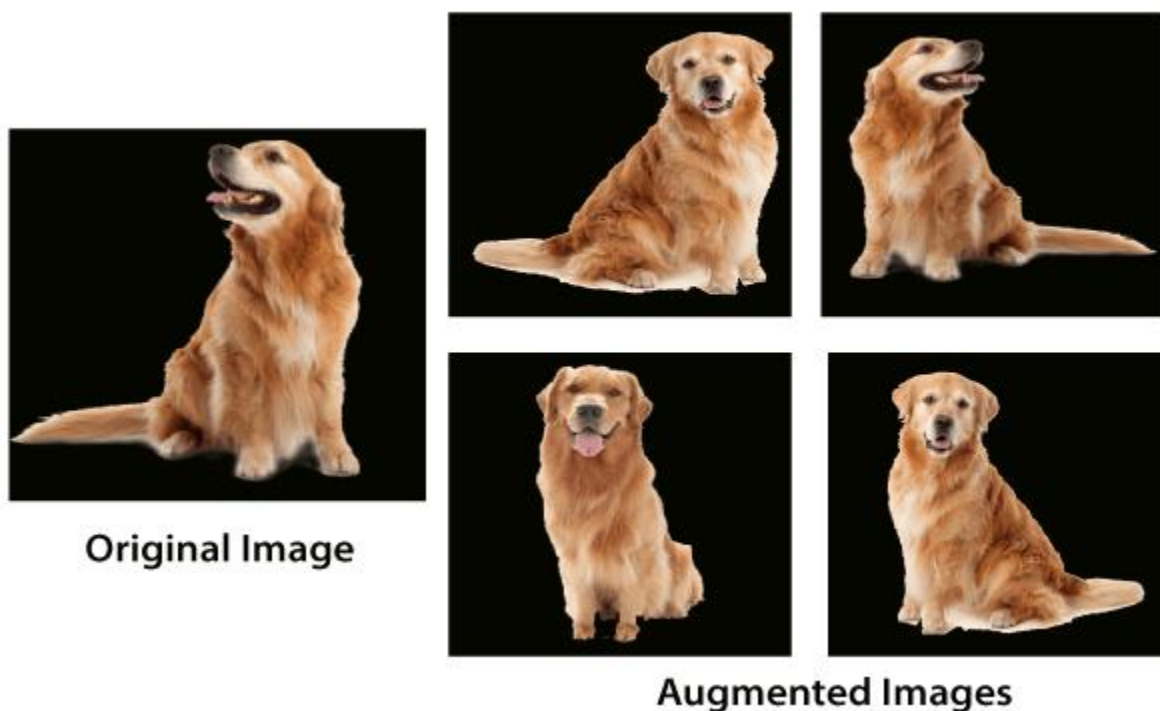


طبق شکل مشاهده میکنیم که از تعداد مشخصی تکرار به بعد ارور داده های تست یا ولیدیشن ما شروع به افزایش میکنند در حالیکه ارور داده های آموزشی همچنان در حال کاهش است. بنابراین مدل دچار مشکل بیش برآزش میشود. برای حل این مشکل ما در جایی که ارور تست شروع به افزایش میکند متوقف میشویم.

### 3- Data Augmentation:

این تکنیک به معنی افزایش سایز داده های آموزشی است. با این تکنیک همانطور که ما داده های بیشتری را اضافه میکنیم مدل در overfit شدن ناتوان میشود و مجبور به تعمیم دهی میشود. تعدادی از aug های معروف تصویر عبارتند از flipping, translation, rotation, scaling, chaning, brightness و همچنین اضافه کردن نویز.

#### Data Augmentation



همانطور که در تصویر مشاهده میکنیم تعداد زیادی تصویر شبیه به تصویر اصلی تولید شده است. با توجه به گفته استاد حتی اگر بخواهد این داده ها را هم حفظ کند. مثلاً همین تصویر سگ را حفظ کند تعداد بیشتری از حالت های مختلف سگ را میبیند و حفظ میکند. در نتیجه خطای تست کمتر میشود.

#### 4- استفاده از regularization:

این تکنیکی است که پیچیدگی مدل را کاهش میدهد و اینکار را با اضافه کردن ترم جریمه به تابع ضرر انجام میدهد. ظرفیت مدل رو محدود میکند و اجازه نمیدهد وزن های شبکه هر چیزی باشند. دو تا از تکنیک های معمول آن L1 و L2 هستند.

L1: کاهش جمع قدر مطلق وزن ها

$$L(x, y) \equiv \sum_{i=1}^n (y_i - h_{\theta}(x_i))^2 + \lambda \sum_{i=1}^n \theta_i^2$$

L2: جمع مربع وزن ها

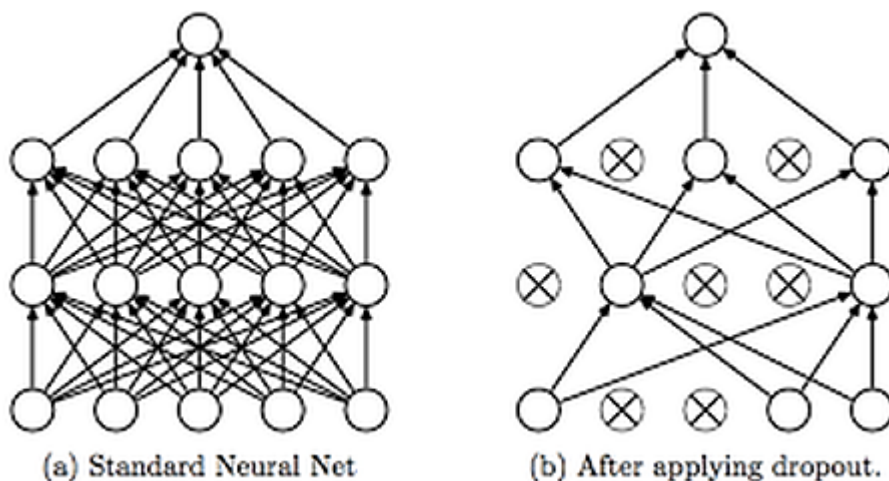
$$L(x, y) \equiv \sum_{i=1}^n (y_i - h_{\theta}(x_i))^2 + \lambda \sum_{i=1}^n \theta_i^2$$

نرم 1 سعی میکند W های زیادی را صفر کند درحالیکه که نرم 2 سعی میکند همه W ها را کوچک کند.

L1 Regularization	L2 Regularization
1. L1 penalizes sum of absolute values of weights.	1. L2 penalizes sum of square values of weights.
2. L1 generates model that is simple and interpretable.	2. L2 regularization is able to learn complex data patterns.
3. L1 is robust to outliers.	3. L2 is not robust to outliers.

5- استفاده از dropouts: این تکنیک شبکه را تغییر میدهد و در هر تکرار به طور تصادفی نورون ها را حذف میکند چون شبکه نباید به تک نورون حساس باشد یعنی اگر چه بود، چه نبود، باید بتواند خوب تصمیم گیری کند و اینطوری رباست تر میشود. وقتی ما مجموعه مختلفی از نورون ها را حذف میکنیم برابر با این است که شبکه های عصبی مختلف را به صورت همزمان آموزش میدهیم که میتوانند همدیگر را تقویت کنند(ensemble).

نویز ما اینجا ضرب شونده ست. نورون ما یا در 0 ضرب میشود یا در  $1/p$ . پس dropout هم یک نوع اضافه کردن نویز به مدل میباشد. (از نسخه inverted dropout)



ثابت شده است که این تکنیک بیش برآزش را در مسئله های متنوعی مانند image classification، image segmentation، word embedding و semantic matching کاهش میدهد.

منابع:

- 1- <https://www.kdnuggets.com/2019/12/5-techniques-prevent-overfitting-neural-networks.html>

2- جلسه کلاس درس