1. Overfitting به مدلی اشاره دارد که داده های آموزشی را خیلی خوب مدل می کند. تطبیق بیش از حد زمانی اتفاق می‌افتد که یک مدل جزئیات و نویز موجود در داده‌های آموزشی را تا حدی بیاموزد که بر عملکرد مدل در داده‌های جدید تأثیر منفی بگذارد. این بدان معنی است که نویز یا نوسانات تصادفی در داده های آموزشی به عنوان مفاهیم توسط مدل انتخاب شده و یاد می شود. مشکل این است که این مفاهیم برای داده‌های جدید اعمال نمی‌شوند و بر توانایی تعمیم مدل تأثیر منفی می‌گذارند.

وقتی می بینید که مدل روی داده های آموزشی عملکرد خوبی دارد اما در داده های ارزیابی عملکرد خوبی ندارد، مدل شما بیش از حد به داده های آموزشی شما برازش می کند. این به این دلیل است که مدل داده‌هایی را که دیده است به خاطر می‌سپارد و نمی‌تواند به نمونه‌های دیده نشده تعمیم دهد.

تناسب بیش از حد در مدل تنها زمانی قابل تشخیص است که داده ها را آزمایش کنید. برای تشخیص مشکل، می‌توانیم تقسیم Train/test را انجام دهیم.

در تقسیم آزمون قطار مجموعه داده، ما می توانیم مجموعه داده خود را به مجموعه داده های آزمایشی و آموزشی تصادفی تقسیم کنیم. ما مدل را با مجموعه داده آموزشی آموزش می دهیم که حدود 80٪ از کل مجموعه داده است. پس از آموزش مدل، آن را با مجموعه داده آزمایشی آزمایش می کنیم که 20 درصد کل مجموعه داده است.

حال، اگر مدل با مجموعه داده آموزشی خوب عمل کند اما با مجموعه داده آزمایشی عملکرد خوبی نداشته باشد، احتمالاً مشکل بیش از حد برازش دارد.

به عنوان مثال، اگر مدل دقت 85٪ را با داده های آموزشی و 50٪ دقت را با مجموعه داده های آزمایشی نشان دهد، به این معنی است که مدل عملکرد خوبی ندارد.

راهکارهای جلوگیری:

- Train with more data
- Data augmentation
- Addition of noise to the input data
- Feature selection
- Cross-validation
- Simplify data
- Regularization
- Ensemblin

هدف ما رسیدن به مدلی است که واریانس ان روی داده ها کم باشد و همینطور میانگین انها هم نزدیک به برچسب های ما باشد.

شکل سمت راست حالتی است که ما واریانس بالایی داریم و مدل ما دقیقا به صورت کامل توانسته داده های مارا تفکیک کند ولی میدانیم هدف ما رسیدن به نتیجه خوب در داده های تست است و اصولا داده های تست ما به شکل های متفاوت تری نسب به داده های تست ما هستند و عملکرد انها روی داده هایی که تا حالا مدل ما ندیده است خوب نخواهد شد.

هنگامی که یک مدل با داده های زیادی آموزش می بیند، شروع به یادگیری از نویز و ورودی داده های نادرست در مجموعه داده های ما می کند. و هنگام آزمایش با داده های آزمون، واریانس بالا را نشان می دهد. سپس مدل به دلیل جزئیات و نویز زیاد، داده

ها را به درستی دسته بندی نمی کند. دلایل اضافه برازش روش های غیر پارامتری و غیر خطی هستند زیرا این نوع الگوریتم های یادگیری ماشین آزادی بیشتری در ساخت مدل بر اساس مجموعه داده دارند و بنابراین واقعاً می توانند مدل های غیر واقعی بسازند.

2. به ازای x=1 سوال را حل میکنیم:

در ابتدا شبکه را forward میکنیم :

$$Y = 20, \quad X_1 = 1, \quad W_1 = 1, \quad X_2 = 1, \quad W_2 = 2, \quad X_3 = 1, \quad W_3 = 3, \quad X_4 = 1, \quad W_4 = -2, \quad X_5 = 1, \quad W_5 = -1,$$

$$Y_{hold} = X_1 * W_1 + \cdots + X_5 * W_5 = 3$$

$$linear\ (Y_{hold}) = Y_{pred} = 3$$

حال بیایم مقدار ارور را باتوجه‌به فرمول زیر حساب کنیم:

$$Ei = \frac{(Y - Y_{pred})^2}{2}$$

که در فرمول بالا مقدار ارور برای y، i ام حساب شده است.out(yi) برابر با مقدار خروجی yi است و همچنین در فرمول بالا Yi مقداری است که ما انتظار داشتیم به‌عنوان خروجی بگیریم.

$$Ei = \frac{(20 - 3)^2}{2} = 144.5$$

حال در مرحله بعدی نیاز داریم به‌صورت backward حرکت کنیم و وزن‌ها را اصلاح کنیم.

$$\frac{\partial Error}{\partial w1} = \frac{\partial Error}{\partial Y} * \frac{\partial Y}{\partial Y_{pred}} * \frac{\partial Y_{pred}}{\partial w1}$$

$$\frac{\partial Error}{\partial Y} = 2 * \frac{1}{2} * \left(Y - Y_{pred}\right) * (-1) = (20 - 3) = 17$$

$$\frac{\partial Y}{\partial Y_{pred}} = \frac{\partial S(Y_{pred})}{\partial Y_{pred}} = 1$$

$$\frac{\partial Y_{pred}}{\partial w1} = X_1 = 1$$

$$\frac{\partial Error}{\partial w1} = 17 * 1 = 17$$

$$New\ w1 = (\mathbf{1 - 0.09}) * \mathbf{1} - 0.1 * 17$$

$$\eta\ (learning\ rete) = 0.1$$

$$\boldsymbol{New\ w1 = -0.79}$$

$$\frac{\partial Error}{\partial w2} = 17 * 2 = 34$$

$$New\ w1\ = (\mathbf{1} - \mathbf{0.09}) * \mathbf{1}\ -\ 0.1\ * 34$$

$$\boldsymbol{New\ w2} = \textbf{-2.49}$$

$$\frac{\partial Error}{\partial w3} = \ 17 * 3 = 51$$

$$New\ w1\ = (\mathbf{1} - \mathbf{0.09}) * \mathbf{1}\ -\ 0.1\ *\ 51$$

$$\boldsymbol{New\ w3} = \textbf{-4.19}$$

$$\frac{\partial Error}{\partial w4} = \ 17 * (-2) = -34$$

$$New\ w1\ = (\mathbf{1} - \mathbf{0.09}) * (\mathbf{1}) +\ 0.1\ *\ 34$$

$$\boldsymbol{New\ w4} = \textbf{4.31}$$

$$\frac{\partial Error}{\partial w5} = \ 17 * (-1) = -17$$

$$New\ w1\ = (\mathbf{1} - \mathbf{0.09}) * (\mathbf{1}) +\ 0.1\ *\ 17$$

$$\boldsymbol{New\ w5} = \textbf{2.61}$$

3. نتیجه به دست امده در حالت پایه:

```
loss: 0.0015 - accuracy: 1.0000 - val_loss: 1.6969 - val_accuracy: 0.6000
```

```
Test loss: 1.9180108308792114
Test accuracy: 0.5600000023841858
```

در این مرحله ما به طور یقین ***overfit*** شده این و دلیل ان هم رسیدن به دقت 100 درصد در داده های اموزش است در حالی که در داده های ***val*** این مقدار 60 درصد و در داده های تست به 56 درصد رسیده است و مدل ما تنها داده های اموزش را یاد گرفته است.

در حالتی که ***augmaatation*** را به صورت دستی محاسبه کردیم :

```
loss: 0.2582 - accuracy: 0.8846 - val_loss: 1.3645 - val_accuracy: 0.5882
```

```
Test loss: 1.4051415920257568
Test accuracy: 0.6054902076721191
```

در این مرحله ما مدل به به طور کامل نیمشه گفت ***overft*** شده این و دلیل ان هم رسیدن به دقت 88 درصد در داده های اموزش است در حالی که در داده های ***val*** این مقدار 60 درصد و در داده های تست به***60*** درصد رسیده است و مدل ما تنها داده های اموزش را یاد گرفته است.

در حالت استفاده از *ImageDataGenerator* ما هم با *MLP* و هم با *CNN* توانستیم به دقت 100 درصد در داده های اموزش و داده های تست برسیم که بهترین نتیجه ممکن است.

البته میتوان گفت در اینجا هم روی داده های اموزش، *overfit* شده ایم.

3. با استفاده از *L2, Dropout, KFold* و همینطور کمتر کردن ایپکا های اموزش توانستیم به دقت برسیم در جالی که بدون این موارد به 80 درصد رسیده بودیم.

دلیل *overfit* بودن روی داده های تست این است که در ابتدا ما به دقت 100 درصد روی داده های اموزش رسیدیم و دقت ما روی داده های *val* ثابت رو 66 مانده بود.

در این تکنیک، آموزش قبل از اینکه مدل شروع به یادگیری نویز درون مدل کند، متوقف می‌شود. در این فرآیند ضمن آموزش مدل به صورت تکراری، پس از هر تکرار، عملکرد مدل را اندازه گیری کنید. تا تعداد مشخصی از تکرارها را ادامه دهید تا زمانی که یک تکرار جدید عملکرد مدل را بهبود بخشد.پس از آن نقطه، مدل شروع به اضافه کردن داده های آموزشی می کند. از این رو ما باید قبل از اینکه یادگیرنده از آن نقطه عبور کند، فرآیند را متوقف کنیم.توقف فرآیند آموزش قبل از اینکه مدل شروع به گرفتن نویز از داده ها کند به عنوان توقف اولیه شناخته می شود.

اعتبارسنجی متقاطع یکی از تکنیک های قدرتمند برای جلوگیری از برازش بیش از حد است. در تکنیک اعتبارسنجی متقاطع *k-fold* کلی، مجموعه داده را به زیرمجموعه های داده با اندازه *k* برابر تقسیم کردیم. این زیرمجموعه ها به نام *folds* شناخته می شوند.

اگر زمانی که یک مدل پیچیده است، بیش از حد برازش اتفاق می‌افتد، می‌توانیم تعداد ویژگی‌ها را کاهش دهیم. با این حال، بیش از حد ممکن است با یک مدل ساده تر، به طور خاص تر مدل *Linear* نیز رخ دهد، و برای چنین مواردی، تکنیک های منظم سازی بسیار مفید هستند.منظم سازی محبوب ترین تکنیک برای جلوگیری از برازش بیش از حد است. این گروهی از روش‌ها است که الگوریتم‌های یادگیری را مجبور می‌کند یک مدل را ساده‌تر کنند. استفاده از تکنیک منظم سازی ممکن است کمی سوگیری را افزایش دهد اما کمی واریانس را کاهش دهد. در این تکنیک، تابع هدف را با اضافه کردن عبارت جریمه، که با یک مدل پیچیده‌تر ارزش بالاتری دارد، اصلاح می‌کنیم.

منابع:

[Overfitting in Machine Learning - Javatpoint]

https://stackoverflow.com/questions/30230592/loading-all-images-using-imread-from-a-given-folder