

به نام خدا

تابع ضرر (loss)

برای آموزش یک شبکه عصبی، باید یک تابع از دست دادن تعریف کنیم. این معیاری است که نشان می دهد شبکه عصبی ما چقدر با کامل بودن فاصله دارد. وقتی شبکه عصبی را آموزش می دهیم، یک تابع ضرر را بهینه می کنیم.

ما از آنتروپی متقاطع به عنوان تابع ضرر استفاده خواهیم کرد. آنتروپی متقاطع را به صورت زیر محاسبه می کنیم.

$$\text{cross entropy} = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{if } y = 0 \end{cases}$$

ما مقادیر آنتروپی متقاطع را برای تمام نقاط داده با هم ضرب می کنیم.

فرض کنید ما دو مدل برای مقایسه روی یک مجموعه داده کوچک با 4 نقطه داده داریم. در اینجا جدولی از مقادیر واقعی، احتمالات پیش بینی شده برای مدل 1 و احتمالات پیش بینی شده برای مدل 2 آورده شده است.

Target	Model 1 Prediction	Model 2 Prediction
1	0.6	0.5
1	0.8	0.9
0	0.3	0.1
0	0.4	0.5

For model 1 →

$$0.6 * 0.8 * (1 - 0.3) * (1 - 0.4) = 0.2016$$

For model 2 →

$$0.5 * 0.9 * (1 - 0.1) * (1 - 0.5) = 0.2025$$

***هر چه مدل بهتر باشد آنتروپی متقاطع بالاتر خواهد بود، بنابراین از آنجایی که مدل 2 آنتروپی متقاطع بالاتری نسبت به مدل 1 دارد، مدل بهتری است.

Backpropagation

یک شبکه عصبی دارای پارامترهای زیادی است که ما می توانیم آنها را کنترل کنیم. برای هر گره چندین ضریب وجود دارد و می تواند تعداد زیادی گره وجود داشته باشد! فرآیند به روز رسانی این مقادیر برای همگرایی در بهترین مدل ممکن بسیار پیچیده است. شبکه عصبی به سمت عقب از گره خروجی کار می کند و به طور مکرر ضرایب گره ها را به روز می کند. به این فرآیند حرکت به سمت عقب در شبکه عصبی، پس انتشار یا پس پاپ می گویند.

ما در اینجا تمام جزئیات را بررسی نمی کنیم زیرا شامل محاسبه مشتقات جزئی می شود، اما ایده این است که همه مقادیر ضرایب را مقداردهی اولیه کنیم و مقادیر را به طور تکراری تغییر دهیم تا در هر

تکرار شاهد بهبود در تابع ضرر باشیم. در نهایت ما دیگر نمی توانیم تابع ضرر را بهبود ببخشیم و سپس مدل بهینه خود را پیدا کرده ایم.

قبل از ایجاد یک شبکه عصبی، تعداد گره ها و تعداد لایه ها را ثابت می کنیم. سپس از `backprop` برای به روز رسانی مکرر تمام مقادیر ضرایب استفاده می کنیم تا زمانی که در یک شبکه عصبی بهینه همگرا شویم.