**Introduction to TensorFlow**

من سیاوش کاوسی هستم و می خواهم TensorFlow library بهتون معرفی کنم.

خب برای شروع بهتره که ببینیم یادگیری ماشین چیه؟!! و بعد میریم سراغ اصل موضوع ☺

یادگیری ماشین یک شاخه از هوش مصنوعیه که کامپیوتر را قادر میسازه که بدون اینکه برنامه نویسی بشوند یاد بگیرند!

حالا این یعنی چی؟!! بریم بیشتر راجع بهش بحث کنیم تا به یک مفهوم کلی ازش برسیم

فرض کنید که ما می خواهیم با استفاده از یادگیری ماشین و داشتن یک مجموعه داده(dataset) که شامل 50000 تا عکس با 10 برچسب(label) است، عکس هایی که بهمون نشون میدن و شبیه عکس هایی است که در dataset مون هست را حدس بزنیم که شبیه کدوم یکیه (کدوم label)

ساده ترین راه اینه که ما بیاییم عکس موردنظر را با تک تک عکس های dataset، پیکسل پیکسل مقایسه کنیم و جمع اختلاف پیکسل های هر کدوم که کمتر شد (یعنی نزدیک ترین همسایه) میشه حدسمون

اما به نظر میاد که این روش کار در بیشتر مواقع ناکارآمده مخصوصا وقتی که تعداد داده ها بیشتر بشه...

روش بهتر اینه که ما با استفاده از داده هایی که داریم (همین 10000 تا عکس) یک سری پارامتر رو train کنیم مثلا طبق نمودار اگر فرض کنید که اون عکس های dataset مون، x های موجود در نمودار باشند برای حدس زدن میتوانیم یک خط بکشیم و این میشه تابع حدسمون!!! که به انجام این فرآیند(process) میگیم forward propagation ☺

حالا مرحله بعد اینه که پارامترهایی که در نظر گرفتیم را مینیمم کنیم که همونطور که در شکل میبینید اگر پارامتر ما x باشد دو راه برای رسیدن به نقطه مینیمم داریم که راه عددیه و راه دیگه استفاده از مشتق(differentiation in calculus) که چون راه اول computational intensive از مشتق گرفتن استفاده می کنیم که در یادگیری ماشین و به خصوص neural network ها بهش میگیم backward propagation ☺

اینا هم میشه حالات پیچیده تر گراف ساده اول

اوکی حالا میریم سراغ اصل مطلب؛ **TensorFlow**

Library ایه که توسط گوگل توسعه داده شده و در بخش های مختلف خودش هم ازش استفاده میشه

حالا شاید از خودتون بپرسین اصن چرا TensorFlow؟!! اول از همه به خاطر اینکه از زبون Python که در دنیای scientist ها رایجه استفاده میکنه و ویژگی های دیگری مثه Auto-Differentiation ، Portability داره که حالا جلوتر میگم که چین! و نکته مثبت و مهمش هم اینه که همه کارای computational intensive را به صورت بهینه در C++ انجام میده و فقط نقطه شروع و پایان در Python اههههه... در واقع باید کل محاسبات که به صورت یک گراف هست رو بکشیم و بعد اجرا کنیم

حالا بهتره با چند تا اصطلاح هم آشنا بشیم

مثلا در این library به جریان داده ها(data flow) میگیم tensors و به گره ها هم عملیات(ops) میگیم

پارامترهایی که آپدیت میشن رو variable در نظر می گیریم و داده هایی که باید به گراف داده بشن که پردازش بشن رو درون placeholder قرار میدیم یعنی یه placeholder تعریف میکنیم که بتونیم داده با سایز متغیر رو به گراف تزریق کنیم

بعد این گراف رو در یک session اجرا می کنیم

همین ☺

تشکرات ☺