

Pytorch pensum

Pytorch

Pytorch er et gratis open-source machine-learning **framework** til python.

Et framework er generelt større end et library, og kan stille krav til fil- og kodenstruktur. Det kan gøre det sværere at komme i gang med, da der kan være en specifik syntaks man skal følge. Derfor har nogle frameworks genereret **boiler plate** kode, hvor brugeren kan udfylde en skabelon der følger frameworkets konventioner. Der kan være fordele og ulemper ved boiler plate. Det kan være svært at vide hvordan et program bliver eksekveret og hvis man kommer til at fjerne en del af skabelonen kan programmet risikere at fejle.

Husk stadig at følge generelle principper for god kodeskik, selvom I arbejder indenfor et framework.

Et pytorch-program består typisk af tre dele: et neuralt netværk-module, et træningsloop samt et main-script.

NN module

Det neurale netværk-modul er en python klasse, men i pytorch har man valgt at kalde det et modul. Denne klasse består af to ting: en init-funktion der instantierer pytorch's hovedklasse ([torch.nn.Module](#)) og opsætter det neurale netværk, samt en forward-funktion som bliver kaldt under træning og evaluering til at bevæge data igennem netværket. Klassen må gerne have flere funktioner, hvis man tænker at det er relevant.

Træningsloop

Selve træningen af en model kan variere meget alt efter hvilken algoritme man bruger. De følger dog samme princip. En mængde data bliver kørt igennem det neurale netværk-modul. Modulet returnerer en værdi som bliver givet til en loss-funktion, som udregner en værdi der repræsenterer præcisionen i modellen. Herefter optimeres modellen. Hvordan modellen bliver optimeret varierer meget imellem algoritmer. Det kunne være ved backpropagation eller lignende.

Træning foregår over et predefineret antal epochs eller indtil man har opnået en ønsket præcision. Typisk vil man definere begge, da det ikke er sikkert, at man kan nå den ønskede præcision i et realistisk tidsrum.

I reinforcement learning vil der også være et såkaldt test-loop/evaluerings-loop, hvor man kører et separat datasæt igennem modulet, for at se hvordan systemet håndterer data det aldrig har set før.

Vi foreslår at man sætter trænings-loop og lignende ind i en klasse for sig selv.

Main-scriptet

I main-scriptet vil man sætte systemet igang. Det vil sige at man instantierer sit NN module, enten ved at instantiere en ny, eller loade en gemt model. NN-modulet sørger for at hente eventuelle konfigurationsfiler, og for at programmet køre i den rigtige rækkefølge. Det sørger også for at gemme modellen efter træning, hvis man ønsker det.