

Applicerad AI Inlämning 3

Teknikhögskolan

January 2022

1 Generell information

Det går utmärkt att utföra uppgifterna i andra ramverk än PyTorch (t.ex. Keras eller Tensorflow)

2 Regression

2.1

För G

Ladda in regressionsdatan och dela upp den i ett tränings-, test- och evalueringsset. Gör ett Feedforwardnätverk med följande lager:

- Linear Layer 1, med dimensioner (n_features, 32) med efterföljande ReLU
- Linear Layer 2, med dimensioner (32, 16) med efterföljande ReLU
- Linear Layer 3, med dimensioner (16, 1)

Använd lämplig lossfunktion för regressionsmodellen. Kör som mest 30 epoker av träning.

Gör följande plots:

- Line plot för training lossen för varje epok
- Scatter plot för predictions på test setet mot faktiska värden när modellen är färdigtränad

För VG

Ladda in datan till ett Dataset och DataLoader. Använd dessa för att göra batches med batch size 8. Se till att använda GPU för beräkningarna - detta finns tillgängligt på Google Colab om man inte själv äger ett kompatibelt GPU.

Lägg till ett "dropout-layer" (`nn.Dropout`) mellan Layer 1 och Layer 2, efter den andra ReLU-aktiveringen. **OBS!** I evalueringen måste `nn.Dropout` stängas av (kan göras med `model.eval()` innan evaluering).

För varje epok (forward- och backwardpass av hela träningsdatan) - spara historik för bästa performance i termer av evaluation loss på ditt evalueringsset. Om det är bättre - spara ner parametrarna till modellen med `torch.save` (Länk).

Sätt även som kriterie för träningen att den ska stanna efter att evaluation loss inte har förbättrats på de senaste 3 epokerna eller efter 30 epoker.

Gör sedan en metod som laddar in dina tränade vikter/värden för modellen och verifiera att det fungerar genom att testa performance på evaluerings- och testset.

3 Klassificering

Ladda in `torchvisions` MNIST Fashion set (Länk). Dela upp training set till train och validation set.

För G

Inspektera mängden labels och avgör vilken dimension som krävs för output i det sista lagret. Använd ett feedforwardnätverk **eller** valfri annan arkitektur.

Använd för ett feedforwardnätverk max 3 hidden layers med ReLU som aktiveringsfunktion. Avgör vilken lossfunktion som lämpar sig för ett klassificeringsproblem.

Använd accuracy som score för att avgöra performance på ditt testset.

För VG

Jämför training/evalueringsloss över tid för två olika optimizers - Adam och SGD. Testa även att lägga till en learning rate scheduler i din träningsloop (Länk)

Plotta en confusion matrix för predictions av ditt testset.

Spara dina tränade vikter med `torch.save` på samma sätt som i föregående uppgift. Gör även här en metod som kan ladda in dina sparade vikter och göra predictions utan att träna om.

Jämför tiden det tar för träningen för batch size = [2, 4, 8, 16, 32]. Växer den linjärt?