

Regularization & Dropout

Övningssession

12/06 – 2024

- Dropout

Observera följande kodsnuitt.

```
dropout_layer = nn.Dropout(p=0.2)
input = torch.randn(1,10)
output = dropout_layer(input)
```

- 1) Vad tror du koden gör?
- 2) Vad får du för output om du kör koden ett par gånger? Does it make sense?
- 3) Anta att vi ändrar batch_size till 3. Hur ändrar vi i koden ovan?
- 4) Ändra batch_size till 3, och kör koden ett par gånger. Vad får du för output? Does it make sense?

- Regularisering

Gå återigen in på [följande interaktiva sida](#) där du kan bygga & träna Neurala Nätverk.

- 1) Försök atts kapa en situation där modellen du tränar uppenbarligen, och tydligt, överanpassar sig på träningsdatan. *Tips: använd mycket hög noise!*
- 2) När du tränat färdigt, scimma över alla connections mellan neuronerna och scimma igenom ungefär vilken storleksordning alla vikter ligger på.
- 3) Träna nu om samma nätverk som innan, på samma data, igen. Men den här gången lägger du till regularisering. Lokalisera och lägg till L1 regularisering.
- 4) Testa att träna med varje värde på lambda ett par gånger, och jämför resulterande train & test loss. Does it make sense?
- 5) När du tränat ett regulariserat nätverk, som förhoppnings överanpassat sig mindre än tidigare, scimma återigen över alla vikter och notera deras storlekar. Vad märker du?

- Träning i PyTorch

Välj nu ett enkelt träningsdataset som du jobbat med tidigare, och testa nu att träna ett Neurtalt Nätverk med regularisering och/eller dropout layers.

Försök att träna större och större nätverk och (förhoppningsvis) overfit på din data. Lägg därtill på

1. Regularisering
2. Dropout (olika lager, olika rates, testa!)
3. Både Regularisering och Dropout