Datorlaboration 4

Josef Wilzén

14 september 2021

Allmänt

Datorlaborationerna kräver att ni har R och Rstudio installerat.

- Kodmanual: länk
- asgf
- Dataset till vissa uppgifter finns här.
- **ISL**: An Introduction to Statistical Learning,
  - Boken: länk
  - R-kod till labbar: länk
  - Dataset: länk och länk
- **IDM**: Introduction to Data Mining
  - Kod till boken finns härhär
  - Sample chapters
- Dataset till vissa uppgifter finns här. mnist data finns på Lisam

Notera att ni inte behöver göra alla delar på alla uppgifter. Det viktiga är att ni får en förståelse för de olika principerna och modellerna som avhandlats. Dessa uppgifter ska inte lämnas in, utan är till för er övning.

Datauppdelning

För att motverka överanpassning bör ni dela upp data till träning-, validering-, (och testmängd). Detta kan göras med `createDataPartition()` från `caret`-paketet. Argument till den funktionen som är av vikt här är p som hur stor andel av observationerna som ska användas till träningsmängden. Ni kan också använda `subset()` för att göra detta också, men det blir svårare att tydligt ange de observationer som ska tilldelas till valideringsmängden. Denna uppdelning ska ske slumpmässigt. Notera att om en testmängd ska skapas måste uppdelningen ske en gång till från valideringsmängden.

Keras

- Se CHEAT SHEET för Keras. Dokumentation för keras finns här

Del 1: Optimiering av neurala nätverk

Här kommer vi utgå från Fashion MNIST-data, kolla här. Läs in datamaterialet och se till att ha koll på vad det är för sorts data. Nu ska ni testa olika inställningar på optimeringen.

1. Definera följande modell i keras:

```
> model <- keras_model_sequential()
> model %>%
+ layer_flatten(input_shape = c(28, 28)) %>%
+ layer_dense(units = 128, activation = 'relu') %>%
+ layer_dense(units = 128, activation = 'relu') %>%
+ layer_dense(units = 10, activation = 'softmax')
```

2. Fixera antal epoker till 20. Sätt den globala learning rate till 0.01. Sätt batchstorlek till 128.
3. Hur många parametrar har modellen?

4. Testa nu följande inställningar. Undersök tränings- och valideringsdata under träning och utvärdera på testdata. **Tips**: titta här vid behov.
   (a) Skatta modellen med vanlig SGD
   (b) Skatta modellen med Adam algoritmen. Använd defaultvärden på övriga hyperparametrar.
   (c) Skatta modellen med RMSPROP algoritmen. Använd defaultvärden på övriga hyperparametrar.
   (d) Vilken metod är ni mest nöjd med?
5. Gör om 4) men med batchstorlek 64 och 256. Hur påverkar det optimeringen?
6. Välj batchstorlek 128 och SGD. Ändra learning rate till 1 och 0.0001, vad händer?
7. Upprepa 6) men med Adam eller RMSPROP
8. Välj batchstorlek 128 och SGD. Testa att ändra startvärdena. Detta görs i lagerfunktionerna. Kolla i dokumentationen hur ni ska göra. Testa att låta startvärdena vara väldigt små (men ej exakt noll) och rätt stora. Hur blir resultatet?
9. Välj en metod från 4) och skatta modellen med 40 epoker.
10. Av alla metoder för optimering ni testat, vad tycker ni har funkat bäst? Har det varit stor skillnad?

Del 2: Neurala nätverk regularisering

Kör denna tutorial: länk.

1. IMDB dataset: Ni kommer jobba med ett binärt klassificeringsproblem. För bakgrund på datasetet, se här.
2. Hur många parametrar är det i de baseline model, smaller model och bigger model? Hur relaterar det till modellernas kapacitet?
3. Hur bra tycker ni att L2 regularisering och dropout funkar för baseline model?
4. Utgå från baseline model, testa förändringarna nedan. Utvärdera valideringsdata med plottar lämpliga utvärderingsmått.
   (a) L2 regularisering, men med ett annat värde på hyperparametern
   (b) Kombinera L2 regularisering och dropout
   (c) L1 regularisering, men med minst två olika värden på hyperparametern
   (d) Jämför med en modell utan tidigare nämd regularisering

Del 3: Mer regularisering

Ni ska nu modellera datasetet det simulerade datasetet "NN_reg_data.csv" med neurala nätverk. Datasetet har en prediktor, x och y har ett icke-linjärt samband med y.

- Läs in datamaterialet i R.
- Det finns tre olika y-variabler:
  - `y_true`: det sanna värdet på y
  - `y_less_noise`/`y_more_noise`: `y_true` + olika nivåer av brus.

Ni analyserade detta dataset i förra laborationen. `y_true` är en funktion som varierar mjukt över x variablen. Målet är nu att se om ni kan anpassa en funktion som varierar mer mjukt med hjälp av regularisering.

1. Plotta de olika y mot x. Hur ser sambandet ut?
2. Börja med att ha `y_less_noise` som er responsvariabel.

3. Dela upp data i träning/test (80/20) slumpmässigt

   (a) Skatta neurala nätverk med arkitekturerna nedan. Testa att lägga till olika typer av regularisering. Ni kan ha olika typer av regularisering för de olika modellerna.

      i. Ett gömt lager med 30 noder + Relu
      ii. Ett gömt lager med 100 noder + Relu
      iii. Två gömda lager med 20 + 20 noder +Relu

   (b) Hur presterar modellerna på träning och testdata?

   (c) Jämför med fallet utan regularisering (se era resultat på laboration 3)

   (d) Gör scatterplot med obserade x och y. Lägg till anpassade värden för träning och test (ha olika färger för de olika fallen). Får ni mer mjukt varierande anpassade värden?

4. Dela upp data i träning/test, där de 20% sista observationerna är testmängden (alltså de längst åt höger i era plottar) och resterande i träningsdata.

   (a) Skatta neurala nätverk med arkitekturerna nedan. Testa att lägga till olika typer av regularisering. Ni kan ha olika typer av regularisering för de olika modellerna.

      i. Ett gömt lager med 30 noder + Relu
      ii. Ett gömt lager med 100 noder + Relu
      iii. Två gömda lager med 20 + 20 noder +Relu

   (b) Hur presterar modellerna på träning och testdata?

   (c) Jämför med fallet utan regularisering (se era resultat på laboration 3)

   (d) Gör scatterplot med obserade x och y. Lägg till anpassade värden för träning och test (ha olika färger för de olika fallen). Får ni mer mjukt varierande anpassade värden?

5. Kolla på era resulat i 3) och 4), funkar regulariseringen som ni föreslår? Lyckas ni anpassa en funktion som varierar mer mjukt med hjälp regulariseringen, jämfört med om ni inte har någon regulariseringen?

6. Upprepa 3) och 4) fast använd `y_more_noise`. Nu finns det mer brus i data, så nu blir det viktigare med att ha en lämplig regularisering för att inte överanpassa. Jämför med fallet `y_less_noise`.

7. Tycker ni att neurala nätverk är en lämplig modellklas för detta dataset?

Del 4: Faltade nätverk

Kör denna tutorial: länk. Ni ska här jobba med CIFAR10 datasetet, som är ett klassiskt datasets inom maskininlärning. För mer info,se här och här.

Notera att detta är färgbilder som har tre kanaler (grön, röd, blå) och att beräkningarna blir mycket tyngre.

1. Beroende på hårdvaran i er dator så kan det vara bra att minska ner storleken på datasetet. Testa att börja med hälften av observationerna. Sedan kan ni använda fler om er dator klarar av det.

```
> cifar0 <- dataset_cifar10()
> no_obs<-dim(cifar0$train$y)[1]
> no_obs_test<-dim(cifar0$test$y)[1]
> set.seed(34)
> index<-base::sample(no_obs,size = no_obs/2)
> index2<-base::sample(no_obs_test,size = no_obs_test/2)
> cifar<-cifar0
> cifar$train$x<-cifar0$train$x[index,,,]
> cifar$train$y<-cifar0$train$y[index,]
> cifar$test$x<-cifar0$test$x[index2,,,]
> cifar$test$y<-cifar0$test$y[index2,]
> rm(cifar0)
```

```
> # kolla så att klasserna är hyfsat balanserade:
> table(cifar$train$y)
> table(cifar$test$y)
```

2. Vad är det för sorts data ni jobbar med här? Hur många obs? Hur stora bilder? Vilka klasser finns?
3. Skatta den föreslagna modellen (OBS kan ta lång tid). Hur många parameterar har den?
4. Hur presterar modellen på testdata? Beräkna förväxlingsmatrisen för testdata.
   (a) Vilken klass var lättast att klassificera?
   (b) Vilken klass var svårat?
5. Testa att ändra arkitekturen på modellen.
   (a) Ändra de faltade lagren:
      i. Antal lager, varning: ta ej för många
      ii. Ändra antal och storlek på `filters` i faltade lagren, varning: ta ej för många
      iii. Ändra `pool_size` i pooling-lagren.
      iv. Testa average pooling
   (b) Testa att modifiera de fullt kopplade lagren.
6. Testa att ändra optimeringen
7. Hur presterar den bästa modellen som ni lyckas hitta?
8. Gå in här och se hur de bästa modellerna presterar på CIFAR10 och MNIST.

Del 5: Mer faltade nätverk

Återvänd till Fashion MNIST-data, kolla här. Läs in datamaterialet och se till att ha koll på vad det är för sorts data.

1. Skapa följande modeller:
   (a) Faltat lager + pooling + två fullt kopplade lager
   (b) Faltat lager + pooling + faltat lager + pooling + två fullt kopplade lager
   (c) En egen kombination av lager, men med minst ett faltat lager.
2. Skatta de föreslagna modellerna. Hur många parameterar har de?
3. Hur presterar modellerna på testdata? Beräkna förväxlingsmatrisen för testdata för de olika modellerna.
   (a) Vilken klass var lättast att klassificera?
   (b) Vilken klass var svårat?
4. Om ni vill: testa att lägga till någon regularisering till den bästa modellen i 1). Hur blir prediktionen på testdata?
5. Jämför med er bästa modell med bara fullt kopplade lager på detta dataset. Vilken presterar bäst? Hur stor är skillnaden? Hur skiljer sig antalet parametrar åt?