Datorlaboration 4

Josef Wilzén

September 5, 2024

732G12 Data Mining HT2024

Allmänt

Datorlaborationerna kräver att ni har R och Rstudio installerat.

- Kodmanual: länk
- **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 här
  - 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.

Del 1: Installera Keras

- Installation se här. (**ISL**) och här (posit).
- Är ni i SU-salarna så ska det finnas installerat, se här (finns även på kurshemsidan) för hur ni läser in pakten.
- Se CHEAT SHEET för Keras. Dokumentation för keras finns här och här.

- Utvärdering vid klassificering kan göras med funktionen class_evaluation_keras().

Del 2: Neurala Nätverk: klassificering

Målet med denna övning är att se hur bra neurala nätverk är på att känna igen bilder med siffror.

1. Läs in materialet "mnist_train.csv"[1] som är träningsmängden från MNIST databasen och ange rätt skala på variablerna. Datamaterialet finns under Kursdokument på Lisam. Här lämpar sig `read.csv2()`. Läs också in testmängden "mnist_test.csv" . Tänk på hur keras-paketet vill ha data strukturerat och följ instruktionerna i kodmanualen [2].

2. Skatta ett neuralt nätverk med ett gömt lager och 10 gömda neuroner. Aktiveringsfunktionen ska vara Relu i det gömda lagret och Softmax i outputlagret. I träningsfasen ska ni ni ta bort den så kallade interna validering, alltså att träningsmängden delas upp till en valideringsmängd inuti algoritmen. Detta görs genom att ange `validation_split = 0`. Använd de övriga standardvärden som är angivna i kodmanualen för anpassning. Ta reda på hur många parametrar er modell har.

3. Utvärdera modellen som skattats genom att kontrollera nätverkets anpassningshistorik. Hur många epoker behövdes för att hitta den bästa modellen? Verkar modellen vara nog bra för att modellera datamaterialet? Utifrån träningsmängden vilka siffror verkar modellen ha lyckas sämst med att prediktera? Vilka siffror har modellen predikterat de till istället? Utvärdering vid klassificering kan göras med funktionen class_evaluation_keras().

4. Utvärdera även modellen på er testmängd. Hur ser resultatet ut där? Ta fram förväxlingsmatisen för testdata och jämför med träningsdata.

5. Testa nu att anpassa modellen med intern validering (20 procent) och repetera steg 3) och 4). Hur ser resultatet ut nu? Kan man teoretiskt förvänta sig en bättre modell med denna förändring?

6. Testa nu att skatta en modell med intern validering där ni försöker ändra arkitekturen av nätverket, t.ex. aktiveringsfunktionerna, antalet gömda neuroner, antalet gömda lager osv. Testa några olika modeller. Vad verkar producera bättre resultat?

7. Testa nu att skatta en modell där ni försöker ändra optimeringen av nätverket, t.ex. antalet epoker, batchstorleken, learning rate osv. Vad verkar producera bättre resultat?

[1] mnist-data finns på Lisam.
[2] Se även lab 10.9.2 A Multilayer Network on the MNIST Digit Data i **ISL**

8. Sammanfatta alla dessa modeller som ni skattat och dra slutsatser om huruvida denna sorts data är lämplig att skatta med neurala nätverk och vad som krävs för att få bra resultat.

9. Gå igenom denna tutorial: länk. Här används också MNIST-data, men data läses in med funktionen dataset_mnist(), notera att data har annat format än när ni läste in den i 1), nu är det en array. Dett gör att inputlagret behöver vara annorlunda. Anpassa den föreslagna modellen[3]. Hur presterar den jämfört med er tidigare? Hur många parameterar skattar ni? Förutom själva arkitekturen på nätverket, hittar ni några andra skillnader jämfört med era tidgare modeller?

10. Lab 10.9.2 "A Multilayer Network on the MNIST Digit Data" i **ISL** skattar en liknande modell som ni har gjort i 2) och 9). Läs igenom labben, jämför deras precision för testdata med vad ni fått tidigare.

[3]Den modellen använder layer_dropout(), vi kommer att gå igenom dropout i kursvecka 4.

Del 3: Mer klassificering

Kör denna tutorial: länk.

1. Vad är det för sorts data ni jobbar med här? Hur många obs? Hur stora bilder?
2. Skatta den föreslagna modellen. Hur många parameterar har den?
3. Hur presterar modellen på testdata? Beräkna förväxlingsmatrisen för testdata.
   (a) Vilken klass var lättast att klassificera?
   (b) Vilken klass var svårast?
4. Testa att ändra arkitekturen på modellen.
   (a) Ändra antalet neuroner i lagren
   (b) Testa att lägga till fler lager.
5. Testa att ändra optimeringen
   (a) antalet epoker
   (b) batchstorlekek
6. Hur presterar den bästa modellen som ni lyckas hitta?
7. Se denna video: Parameters vs Hyperparameters

Del 4: Neurala Nätverk: Regression

Kör denna tutorial: länk.

1. Vad är det för data? Hur många förklarande variabler? Hur många observationer?
2. Skatta den föreslagna modellen. Hur många parameterar har den?
   (a) Hur bra blir modellen på testdata?
   (b) Vilken kostnadsfunktion används?
3. Testa att ändra arkitekturen på modellen och utvärdera resultatet på testdata.
4. Testa att ändra optimeringen och utvärdera resultatet på testdata.
5. Skatta en Random forest-modell på samma dataset och utvärdera resultatet på testdata.
   (a) Fungerar Random forest bättre eller sämre än neurala nätverk på dettta dataset?
6. Gå igenom labben 10.9.1 "A Single Layer Network on the Hitters Data" i **ISL**.

Del 5: Mer regression

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

1. Läs in datamaterialet i R.
2. 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.
3. Plotta de olika y mot x. Hur ser sambandet ut?
4. Börja med att ha `y_less_noise` som er responsvariabel.
5. Skatta modellerna nedan på alla observationer.
   (a) Skatta en ploynomregression med `lm()` av ordning 20. Tips: `poly()`
   (b) Skatta en modell med smoothing splines som skattar λ med korsvalidering.
   (c) Skatta[4] neurala nätverk med följande arkitekturer:
      i. Ett gömt lager med 20 noder + Relu
      ii. Ett gömt lager med 200 noder + Relu
      iii. Två gömda lager med 50 + 50 noder +Relu
      iv. Två gömda lager med 100 + 100 noder +Relu
   (d) Gör b), men lägg till polynom av `x` upp till ordning 5 som förklarande variabler till neurala nätverken. Tips: `as.matrix(poly(train_x,degree = 5))`
   (e) Plotta anpassade värden tillsammans med observerad data för de olika modellerna. Hur ser anpassningen ut?
   (f) Ta fram MSE för träning och MSE där ni jämför de anpassade värdena med `y_true`. Hur funkar modellerna?
   (g) Om ni vill: Skatta en trädmodell med cost complexity pruning och/eller k-nearest neighbors och jämför med modellerna ovan.
6. Dela upp data i träning/validering, där de 5% sista observationerna är valideringmängden (alltså de längst åt höger i era plottar) och resterande i träningsdata.
   (a) Gör om stegen i 5), men med den nya uppdelningen.

[4] Välj valfria inställningar på optimeringen. Ex: batch_size=32, epochs = 30, learning_rate = 0.01

7. I 5) försöker ni interpolera en funktion, i 6) försöker ni extrapolera en funktion. Vilken uppgift verkar lättast? Hur fungerade era modeller i de båda fallen?

8. Upprepa 5) och 6) fast använd `y_more_noise`. Hur presterar de olika modellerna när det finns mer burs i data?

9. Skatta några modeller på `y_true` och hela datasetet som träning.

   (a) Skatta modeller med ett gömt lager och:

      i. 5 noder + Relu

      ii. 10 noder + Relu

      iii. Så många noder som krävs för att få en mycket bra anpassning.

   (b) Plotta observerade värden och anpassade värden i samma plot. Hur bra är neurala nätverk på att anpassa en godtycklig funktion?