Machine Learning

Kunskapskontroll 2

Daniel Borgenstedt

EC Utbildning

202403

.

Innehåll

1 Inledning

I denna rapport är tanken att skapa en maskininlärningsmodell för att prediktera MNIST datasetet och för att uppfylla syftet så kommer följande frågeställning(ar) att besvaras:

1. Går det att skapa en modell som kan prediktera MNIST med minst 80%?
2. Kan modellen prediktera utomstående bilder från antingen datorns webkamera eller en mobilkameras bilder?

I denna kunskapskontroll kommer därefter nio utvalda frågor att besvaras längst bak.

1.1 Streamlit

För att kunna besvara frågeställning 2, så skapas en applikation med hjälp av ramverken Streamlit och OpenCV.

2 Teori

2.1 Modeller och dataset

SVC

Support Vector Machine är en algoritm för klassificering och regressionsproblem. För denna modell har ”Linear”, ”Polynomial” och ”RBF” valts som kernel-hyperparametrar vid initial träning. Efter initial träning genomförts så fortsatte testandet med parametersättning för ”C” och ”Gamma”.

SGD Classifier

SGD står för Stochastic Gradient Descent och är en linjär klassificerings algoritm. Stora dataset och komplicerade modeller har nytta av att använda denna modell. Vid initial träning användes standardinställningar.

Logistic Regression

Logistic Regression används för binär klassificering och skapar värden mellan 0 och 1. För initial träning användes standardinställningar.

Dataset

MNIST är det dataset som använts. Datasetet skapades 1994 och innehåller 70 000 bilder på handskrivna enskilda siffror i 28 x 28 format.

3 Metod

3.1 Databas

Open Machine Learning Foundation är en organisation som tillhandahåller över 5000 publika dataset för maskininlärning. Genom att använda Scikit-Learns ramverk ”fetch_openml” laddades MNIST databasen ner för testning.

3.2 Hårdvara

Träningen har gjorts på en Intel CPU 12900 (base clock 3,2GHZ) med 16 cores och 64GB ram minne samt m2 disk med read/write 7000MB/s.

3.3 Testkörning

Vid den första träningen användes samtliga algoritmer med sina standardparametrar för att få en generell uppskattning av vilka modeller som presterade bäst på MNIST datasetet.

Träningsdelarna skalades ner för att lättare kunna klassificeras.

Detta utfördes först på hela datasetet, men då det var extremt tidskrävande (trots bra hårdvara), där vissa modeller tog över 6h att köra, togs beslut om att dela upp det till 20 000 utav de 70 000 bilderna.

Därefter predikterades modellerna på testdelen för att få en uppfattning om vilken som presterade bäst.

3.4 Vald modell

Valet föll på SVC som hade samtliga modeller bättre presterande än SGD och Logistic Regression.

För att kunna få den valda modellen att prestera ytterligare kördes därefter en GridSearch cross validation för hyperparametrarna ”C” och ”Gamma”. Som parameter för GridSearch användes ”scoring = accuracy”.

3.5 Slutlig träning

De parametrarna, som valdes ut som bäst, användes för att träna en ny SVC-modell på fulla datasetet med samtliga 70 000 bilder. Efter modellen var klar sparades denna ner för att kunna öppnas via Streamlit-applikationen.

3.6 Streamlit och OpenCV

Ingen erfarenhet eller kunskap om vare sig Streamlit eller Computer Vision fanns när denna kunskapskontroll startade och de var inte några ämnen som ingick i kursens omfattning. Därför tog det ett par dagar för att kunna läsa in sig kring dessa ämnen och hitta lämpliga förklaringar på vad det var och hur man skulle gå tillväga för att sätta upp dessa.

OpenCV verkar enligt uppfattning var det mjukvarubibliotek som används mest till s.k computer vision, så därför föll valet på att välja det.

Efter att fått upp ett enklare UI i Streamlit för uppladdning av modell samt val av att antingen välja egen webcamera eller att ladda upp egna bilder, så påbörjades tester för att kunna prediktera bilder

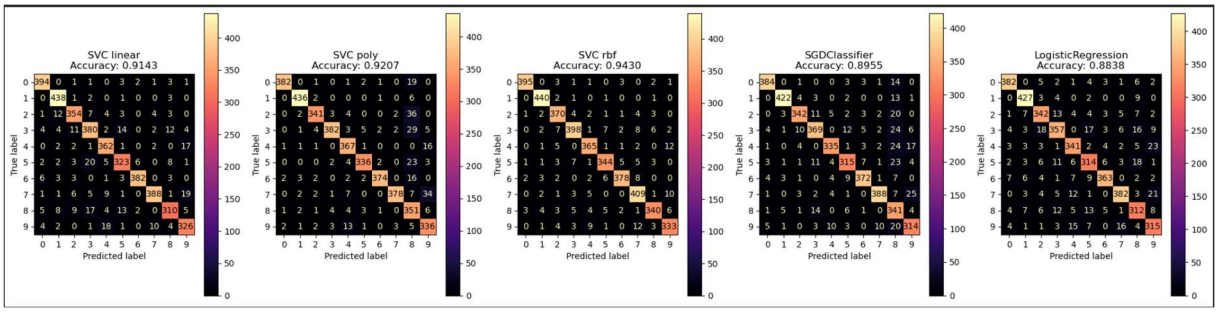
med hjälp av den modell som tränades. Koden skrevs om flera gånger då resultaten inte var framgångsrika till en början.

Bilderna gjordes om till gråskala och därefter ändras färgerna till ren svart och vit via s.k threshold-formatering. Detta för att få en så binär inställning på bilden som möjligt. Slutligen ändrades storleken till att motsvara samma mått som MNIST-bilderna har, 28 x 28 och slutgiltig formatering för att modellen lättare skulle kunna prediktera siffran.

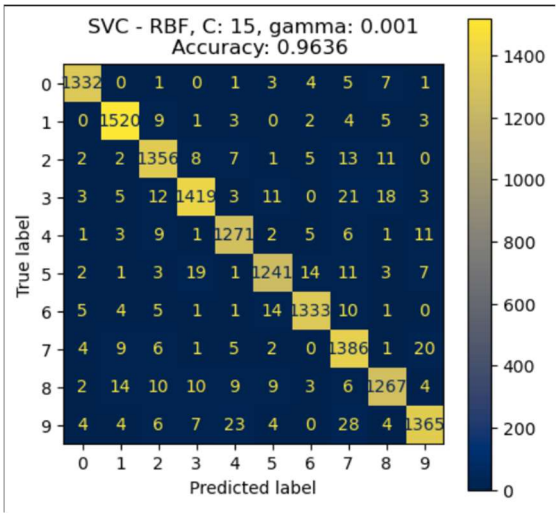
4 Resultat och Diskussion

4.1 Test av modeller

Samtliga SVC modeller presterade bäst i de primära testerna, men det var med relativt liten marginal där det bara skiljde någon procent.



Efter att ha tagit fram de bästa parametrarna för att få hög träffsäkerhet, så landade inställningarna på följande:

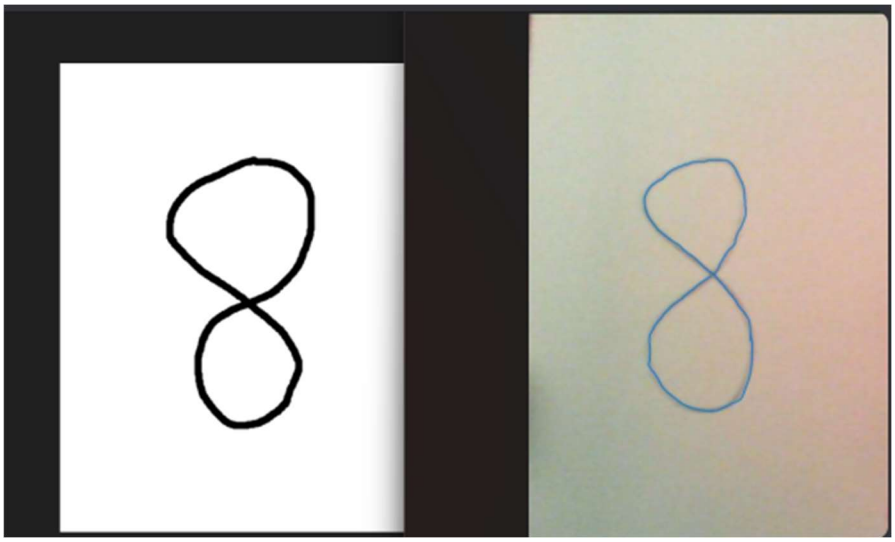


Dock var det även här mycket marginellt. En C-parameter på 5 eller 25 istället för 15 hade bara några få procent sämre resultat. Gamma inställt på 0.001 verkar däremot vara det som påverkade modellen mest. Samtliga topp 5 resultaten visade att det hade större betydelse. En för högt inställd gamma ger en högre komplexitet till en modell och ökar därefter risken för att denna får s.k ”overfitting”.

4.2 Test av egna bilder

I Streamlit-applikation tog det stopp med framgångarna till att börja med. Varje bild som togs och laddades upp fick fel. Koden skrevs om och till slut började några bilder få rätt, men det var fortfarande mycket långt ifrån för att kalla det framgångsrikt.

Efter tre dagar drogs dock slutsatsen att det inte var koden som var felet. Det var istället kamerans upplösning, ljus och inställningar som påverkade resultatet mest. Efter att ha ändrat till att använda kritvit bakgrund med svart tuschpenna och att se till att bilden var så ljus som möjligt med skarp kontrast, så ändrades resultaten, på uppladdade bilder, från att ha rätt på endast 2 av 10 till samtliga 10 siffror korrekt predikterade. Nedanstående vänstra bild predikteras korrekt medan den högra blir fel. När dessa inverterades med thresholding var den vänstra betydligt mer tydlig och därefter också lättare att prediktera.



Webkameran visade sig dock vara för dålig på både ljus och bildkvalité för att bilderna skulle kunna predikteras korrekt.

Tiden det skulle ta att träna alla modeller samtidigt var betydligt längre än vad jag förväntade mig.

I början lät jag det stå över natten, men när det ändå inte var klart på morgonen så drog jag ner på samplingarna. Även med bra hårdvara (för en privatperson) kan det ta lång tid.

5 Slutsatser

Går det att skapa en modell som kan prediktera MNIST med minst 80%?

Svaret på detta är ja. Detta var mycket lättare än jag hade förväntat mig. Jag har varit vilsen i denna kursen då jag inte känt att jag fått ett så bra grepp om maskininlärning som jag hade velat ha.

Detta arbete hjälpte dock till och gav mig lite mer att stå på som grund. Det är lätt att fastna i de matematiska delarna bakom alla olika modeller och att göra kopplingar till vad som är väsentligt.

Det gör att man inte alltid vet vart man ska börja och vilka inställningar som skall göras eller ens vilken modell som passar vart.

Med de ca 96% som SVC producerade, måste jag kalla det för ett mycket glädjande resultat.

Enligt de resultat som presenterats på nätet har dock många kommit fram till ännu bättre resultat genom att använda sig utav CNN och neurala nätverk.

Kan modellen prediktera utomstående bilder från antingen datorns webkamera eller en mobilkameras bilder?

Ja, men det beror väldigt mycket på bildkvalité samt hur bilden processas. Om bilden är ljusklar och tydlig kan det göra stor skillnad.

Kamerans kvalité kan vara mycket avgörande om en bild tagen via den ska kunna predikteras ordentligt. Då mobiler har bättre kameror överlag, är chansen större om en sådan används.

Detta moment var mest komplicerat då vi inte gått in på ämnet om computer vision i kursen. Det tog lång tid innan jag fick grepp om vilken kod som skulle skrivas för att få det att funka eftersom jag fick leta upp den via olika sidor och guider.

Alla gör på olika sätt där med olika ramverk och modeller, vilket inte gör det lättare precis. Innan jag förstod att kvalitén på bilderna hade stor betydelse så famlade jag i mörkret där ett tag och skrev om koden flera gånger utan att förstå vart det verkliga felet låg. Maskininlärning är många gånger en djungel där man inte har en aning om vilken väg man ska gå.

6 Teoretiska frågor

1.**Kalle delar upp sin data i ”Träning”, ”Validering” och ”Test”, vad används respektive del för?**

SVAR: Den delen av det uppdelade datasetet som man ska träna på kallas för ”Träning”. Valideringsdelen används för att utvärdera en viss models utfall med exempelvis inställning av parametrar. Slutligen används Testdelen av datan först när utvärdering är klar i träning och valideringsdel.

2. **Julia delar upp sin data i träning och test. På träningsdatan så tränar hon tre modeller; ”Linjär Regression”, ”Lasso regression” och en ”Random Forest modell”. Hur skall hon välja vilken av de tre modellerna hon skall fortsätta använda när hon inte skapat ett explicit ”valideringsdataset”?**

SVAR: Hon får utvärdera via träningsdatan på de olika modellerna. Någon form av cross validation är lämpligt att använda med randomiserade parametrar.

3**. Vad är ”regressionsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?**

SVAR: Regression används för att utmäta möjliga förhållanden till oberoende variabler såväl som beroende variabler. Man predikterar framtida nivåer/beteenden/priser utifrån de olika faktorerna i datan. Linjär Regression och Logistisk Regression är exempel på olika typer av tekniker som används. Vid t.ex linjär regression mäts punkters avstånd till en rät linje.

4. **Hur kan du tolka RMSE och vad används det till:** $RMSE = \sqrt{n1} \sum_{n1} (y_i - \hat{y}_i)_2$

SVAR: Root Mean Square Error är ett sätt att uppskatta hur bra en maskinmodell predikterar på datan. Ett högt värde visar att det predikterade värdet skiljer sig från det observerade värdet. Det innebär att modellen inte anpassar sig särskilt bra till datan. Men ett för lågt värde kan i sin tur leda till s.k. ”overfitting” av datan.

I Python kan man skriva följande med det ”sanna värdet” och det predikterade:

```
sqrt(mean_squared_error(actual, prediction)
```

5. **Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?**

SVAR: Klassificering är en teknik som predikterar enligt kategorier. Ett exempel kan vara spam-scanning i email. Där klassas ett mail antingen som spam eller inte spam. Det är således binärt, 1 eller 0. Modellen tränas utifrån redan uppdelade förklassificerade exempel.
Decision Tree och KNN är exempel på modeller som används till detta.

En Confusion Matrix är en matris som visar hur många gånger en modell predikterar korrekt och fel. Se exempel i denna rapport.

6. **Vad är K-means modellen för något? Ge ett exempel på vad det kan tillämpas på.**

SVAR: K-means är en kluster-algoritm. Syftet är att lokalisera s.k. centroider, dvs mittpunkten på ett kluster, ett ”mean-värde”. Utifrån centroiderna och dess avstånd till olika punkter, så klassas punkterna till dessa olika kluster. Man väljer antal kluster i förväg när man delar upp datan.

7. **Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding.**

SVAR: Machine learning behöver numerisk data och för att få det finns det olika tekniker.
Om man har kategorisk data som är baserad på text, tex färger, då tilldelas dessa en siffra med Ordinal Encoding. Det blir alltså ['green', 'red', 'blue'] => [0, 1, 2] och nu kan man lättare träna sin data. För One-hot encoding så sätts istället binära värden för varje kategori, dvs 0-1.
[[0, 0, 1]
[0, 1, 0]
[1, 0 ,0]]
Med Dummy encoding så tar man bort en kategori då man kan utesluta den sista när alla föregående inte uppfyller samma kategorisering (Om det inte är någon av red eller green, så är det blue).
[[0, 0, 1] = 'red'
[0, 1, 0]] = 'green'

8. **Göran påstår att datan antingen är "ordinal" eller "nominal". Julia säger att detta måste tolkas. Hon ger ett exempel med att färger såsom {röd, grön, blå} generellt sett inte har någon inbördes ordning (nominal) men om du har en röd skjorta så är du vackrast på festen (ordinal) – vem har rätt?**

SVAR: Båda har korrekt. Rangordning kan tillsättas "manuellt" baserat på preferenser, men det är inte en naturlig ordning. Att den röda skjortan rankas som vackrast är således en egen preferens eftersom en annan person kan tycka att t.ex grön är finast osv.

9. **Vad är Streamlit för något och vad kan det användas till?**

SVAR: Streamlit är ett ramverk som man kan använda för att lätt sätta upp en mini-webserver och skapa appar. Det är Python-baserat och är skapat främst till Data Science relaterade presentationer, men mycket annat kan också naturligtvis göras med det. Mycket kommer redan inbyggt så man slipper göra flera detaljer som man annars gör inom webutveckling som t.ex CSS-hantering m.m.

Appendix A

Top 5 Best Parameters:

1. Parameters: {'C': 15, 'gamma': 0.001, 'kernel': 'rbf'},
   Mean Test Score: 0.9561
2. Parameters: {'C': 5, 'gamma': 0.001, 'kernel': 'rbf'},
   Mean Test Score: 0.9559
3. Parameters: {'C': 25, 'gamma': 0.001, 'kernel': 'rbf'},
   Mean Test Score: 0.9558
4. Parameters: {'C': 50, 'gamma': 0.001, 'kernel': 'rbf'},
   Mean Test Score: 0.9556
5. Parameters: {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'},
   Mean Test Score: 0.9455

Källförteckning

https://www.geeksforgeeks.org/understanding-logistic-regression/

https://www.geeksforgeeks.org/support-vector-machine-algorithm/

https://www.geeksforgeeks.org/what-is-the-difference-between-sgd-classifier-and-the-logistic-regression/

https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/

https://opencv.org/

https://streamlit.io