Machine Learning

En jämförelse av maskininlärningsmodeller för klassificering av MNIST-data

Mahad Elmi

EC Utbildning

Kunskapskontroll

2025-03-20

Abstract

This report explores the application of machine learning models for classifying digit images using the MNIST dataset. The goal was to compare the performance of different models, including Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Random Forest, and Multi-Layer Perceptron (MLP). After training and evaluating the models, the MLP model demonstrated the highest accuracy on both validation and test data. The report also discusses the importance of hyperparameter tuning and data preprocessing in improving the model's performance. Limitations and potential improvements are also addressed, such as enhancing the model's ability to generalize to variations in digit representation.

Förkortningar och Begrepp

SVM = Support Vector Machine
K-NN = k-Nearest Neighbors

RF = Random Forest

MLP = Multi-Layer Perceptron

Följande definitioner är hämtade från Géron (2019):

Hyperparametrar – Parametrar som inte lärs in av modellen utan ställs in manuellt innan träning. Exempel är antalet neuroner i ett dolt lager, inlärningshastighet och antal iterationer.

Förbehandling (Preprocessing) – Förbehandling av data innan träning, till exempel skalning, normalisering och utjämning av data för att förbättra modellens prestanda.

Confusion matrix – En matris som visar antalet korrekta och felaktiga klassificeringar för varje klass. Den hjälper till att analysera modellens prestanda för specifika klasser.

Skapas automatiskt i Word genom att gå till Referenser > Innehållsförteckning.

Innehållsförteckning

1 Inledning

Maskininlärning (ML) har under det senaste decenniet utvecklats till ett av de mest inflytelserika och snabbväxande områdena inom artificiell intelligens (AI). Genom att låta datorer lära sig från data och förbättra sina prestationer utan att explicit programmeras, har ML visat sig vara effektivt inom ett brett spektrum av tillämpningar, inklusive bildigenkänning, naturlig språkbehandling och beslutsfattande (Géron, 2019). En av de mest välkända och etablerade utmaningarna inom bildigenkänning är klassificering av siffror från bild data, vilket ofta betraktas som en grundläggande uppgift inom området datorseende.

Ett av de mest använda dataseten för bildklassificering är MNIST-datasetet (Modified National Institute of Standards and Technology). Datasetet består av 70 000 bilder på siffror (0–9) i storleken 28x28 pixlar och används ofta som en benchmark för att utvärdera och jämföra olika maskininlärningsmodeller (Géron, 2019). Trots att MNIST är ett relativt enkelt datasett jämfört med moderna bildigenkänningsproblem, erbjuder det en värdefull grund för att förstå och analysera olika klassificeringsmetoder.

I detta arbete jämförs fyra olika modeller för bildklassificering av siffror med hjälp av MNIST-datasetet: Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Random Forest (RF) och Multi-Layer Perceptron (MLP). Dessa modeller representerar olika tillvägagångssätt inom maskininlärning från linjära klassificeringsmetoder till ensemble och djupa neurala nätverk. Målet är att undersöka hur modellerna presterar i termer av noggrannhet och generaliseringsförmåga på både validerings- och testdata samt att analysera vilka faktorer som påverkar modellernas prestanda. Genom att jämföra resultaten för de olika modellerna blir det möjligt att identifiera vilken metod som är mest effektiv för denna typ av bildklassificering och varför vissa modeller presterar bättre än andra.

Detta arbete är relevant eftersom förståelse för hur olika maskininlärningsmodeller hanterar bildklassificering kan ge insikter som är tillämpbara på mer komplexa problem inom datorseende. Jämförelsen kan ge vägledning om vilken modell som är mest lämplig för liknande uppgifter samt hur hyperparametrar och dataförbehandling påverkar prestandan.

1.1 Syfte och Frågeställning

Syftet med denna rapport är att jämföra prestandan hos fyra olika maskininlärningsmodeller (SVM, k-NN, Random Forest och MLP) för bildklassificering av siffror med hjälp av MNIST-datasetet. Målet är att identifiera vilken modell som uppnår högst noggrannhet på validerings- och testdata samt att analysera vilka faktorer, såsom hyperparametrar och förbehandling, som påverkar modellernas prestanda.

För att uppfylla syftet kommer följande frågeställningar att besvaras:

Vilken av de fyra modellerna (SVM, k-NN, Random Forest och MLP) uppnår högst noggrannhet på validerings- och testdata?

1. Hur påverkar hyperparametrar och dataförbehandling modellernas prestanda?
2. Vilka styrkor och svagheter har respektive modell vid klassificering av siffror?

2 Teori

2.1 Maskininlärningsmodeller

Vid klassificering av bilddata används olika maskininlärningsmodeller beroende på problemets natur och datans struktur. I detta arbete testades fyra olika modeller: SVM, k-NN, RF, och MLP. Varje modell har sina egna styrkor och svagheter, vilket gör det intressant att jämföra deras prestanda för att identifiera den mest effektiva metoden för bildklassificering av siffror.

2.1.1 Support Vector Machines

SVM är en övervakad maskininlärningsalgoritm som används för klassificering och regression. Modellen fungerar genom att skapa ett hyperplan som separerar klasserna i datan med maximal marginal. Vid linjärt separerbara data fungerar algoritmen genom att skapa ett hyperplan i ett tvådimensionellt eller högre dimensionellt utrymme.

För att hantera icke-linjära data används en så kallad kärntrick (kernel trick), som gör det möjligt att transformera data till ett högre dimensionellt utrymme där det blir linjärt separerbart. I detta arbete användes en SVM-modell med en RBF-kärna (Radial Basis Function) för att hantera komplexa mönster.

2.1.2 k-Nearest Neighbors (k-NN)

k-NN är en enkel men effektiv algoritm som bygger på principen om närhet. Modellen klassificerar en datapunkt baserat på dess k närmaste grannar i datamängden, där klassificeringen sker genom majoritetsröstning från grannarna. En av fördelarna med k-NN är att modellen inte kräver någon träningstid, all beräkning sker vid prediktionstillfället. Nackdelen är dock att modellen kan bli långsam vid stora datamängder eftersom varje prediktion kräver beräkning av avstånd till alla punkter i träningsdatan. Hyperparametern k anger antalet grannar som används för klassificeringen och påverkar modellens prestanda direkt. Ett lågt värde på k gör modellen känsligare för brus, medan ett högt värde gör att modellen blir mer stabil men riskerar att missa finare detaljer i datan. I detta arbete användes k = 5, vilket är en vanlig inställning för att uppnå en balans mellan bias och varians.

2.1.3 Random Forest (RF)

Random Forest är en ensemblemodell som bygger på principen att kombinera flera beslutsträd för att förbättra noggrannheten och minska risken för överanpassning. Modellen fungerar genom att skapa flera beslutsträd under träningen, där varje träd tränas på en slumpmässig delmängd av träningsdatan och med en slumpmässig uppsättning av egenskaper. Klassificeringen sker genom majoritetsröstning från dessa träd, vilket gör modellen mer robust och mindre känslig för variationer i datan jämfört med enskilda beslutsträd.

En viktig hyperparameter i Random Forest är antalet träd (n_estimators). I detta arbete användes 100 träd, vilket är en vanlig inställning som ger en bra balans mellan prestanda och beräkningstid. En annan central hyperparameter är träddjupet (max_depth), som styr hur komplexa de enskilda träden

får bli. Ett för stort djup kan leda till överanpassning, medan ett för grunt djup kan resultera i att modellen inte fångar upp komplexa mönster. I detta arbete användes ett maximalt djup på 15, vilket visade sig ge en bra balans mellan modellens komplexitet och generaliseringsförmåga.

2.1.4 Multi-Layer Perceptron (MLP)

MLP är en typ av artificiellt neuralt nätverk som består av minst tre lager: ett inmatningslager, ett eller flera dolda lager och ett utmatningslager. MLP tillhör kategorin övervakade modeller och används ofta för klassificering och regression på grund av dess förmåga att lära sig komplexa icke-linjära samband i datan.

I detta arbete användes en MLP-modell med ett dolt lager som innehöll 128 neuroner. Detta antal neuroner valdes för att ge en bra balans mellan modellens kapacitet att fånga upp komplexa mönster och risken för överanpassning. Aktiveringsfunktionen ReLU (Rectified Linear Unit) användes i de dolda lagren eftersom den är effektiv för att hantera gradientproblem och möjliggör snabbare konvergens. Modellen tränades med Adam-optimeraren, som kombinerar momentum och adaptiv inlärningshastighet för att förbättra konvergensen. För att minska risken för överanpassning användes en regulariseringsparameter (alpha) inställd på 0,1, vilket visade sig ge en god balans mellan generalisering och prestanda.

2.2 Hyperparametrar

Hyperparametrar är parametrar som inte lärs in av modellen utan ställs in innan träning. Dessa parametrar påverkar modellens struktur och inlärningsförmåga och behöver optimeras för att uppnå bästa möjliga resultat.

Exempel på hyperparametrar i de modeller som användes:

- SVM: Kärntyp (RBF), regulariseringsparameter (C)
- k-NN: Antal grannar (k), avståndsmetod (t.ex. euklidiskt avstånd)
- Random Forest: Antal träd, trädens maximala djup, minsta antal datapunkter för delning
- MLP: Antal neuroner i det dolda lagret, inlärningshastighet, aktiveringsfunktion

För att hitta optimala hyperparametrar kan metoder som GridSearchCV eller RandomizedSearchCV användas för att testa olika kombinationer av inställningar och identifiera den bästa modellen (Géron, 2019).

2.3 Förbehandling av data

Förbehandling av data är ett viktigt steg för att förbättra modellens prestanda. För MNIST-datan normaliserades pixelvärdena till intervallet 0–1 för att undvika problem med gradienter vid träning av MLP. Skalning är särskilt viktigt för modeller som MLP och SVM eftersom dessa modeller är känsliga för värdeskalor och variationer i indata.

Vid träning av modellerna omvandlades varje bild automatiskt från en tvådimensionell matris (28x28 pixlar) till en vektor med 784 element. Detta skedde som en del av modellernas inbyggda databehandling och krävde ingen manuell plattning.

Datan delades också upp i tränings-, validerings- och testdata med proportionerna 80%, 10% respektive 10%. Denna uppdelning gör det möjligt att optimera modellernas prestanda och samtidigt utvärdera deras generaliseringsförmåga på ny data.

3 Metod

Arbetet genomfördes genom att först hämta MNIST-datasetet från OpenML med hjälp av fetch_openml från scikit-learn. MNIST-datasetet innehåller 70 000 bilder i storleken 28x28 pixlar, med motsvarande etiketter mellan 0 och 9.

Förbehandling av datan genomfördes genom att standardisera pixelvärdena med hjälp av StandardScaler från scikit-learn. Detta innebär att varje pixelvärde omvandlades till en z-score genom att subtrahera medelvärdet och dividera med standardavvikelsen. Standardisering är särskilt viktigt för modeller som SVM och MLP eftersom dessa modeller är känsliga för värdeskalor och variationer i indata. Genom att standardisera data förbättras modellens förmåga att konvergera och hitta optimala lösningar.

Datan delades därefter upp i tre separata datasett för träning, validering och testning. Träningsdatan motsvarade 80% av den totala datan, medan de återstående 20% delades jämnt mellan valideringsoch testdatan (10% vardera). Uppdelningen genomfördes med hjälp av train_test_split från scikit-learn med random_state=42 för att säkerställa reproducerbarhet. Efter uppdelningen resulterade datan i följande storlekar:

- Träningsdata: (56 000, 784)
- Valideringsdata: (7 000, 784)
- Testdata: (7 000, 784)

Plattning av bilderna genomfördes automatiskt som en del av scikit-learns hantering av data. Varje bild (28x28 pixlar) omvandlades till en vektor med 784 element innan den matades in i modellerna. Detta gjorde det möjligt att använda bilderna som indata i modellerna utan ytterligare manuell bearbetning.

För att träna och utvärdera modellerna användes fyra olika maskininlärningsmodeller: SVM, k-NN, RF och MLP. Varje modell tränades på träningsdatan och utvärderades först på valideringsdatan för att identifiera den bästa modellen. Slutligen testades modellen på testdatan för att utvärdera modellens generaliseringsförmåga.

4 Resultat och Diskussion

Resultaten från utvärderingen visar att Multi-Layer Perceptron (MLP) presterade bäst med en träffsäkerhet på 97,8 % på testdatan. MLP-modellen hade även en hög träffsäkerhet på träningsdatan (99,9 %), vilket visar att modellen kunde identifiera komplexa mönster i datan på ett effektivt sätt. SVM-modellen visade också starka resultat med en testnoggrannhet på 96,6 %, vilket tyder på att modellen kunde hantera komplexiteten i datan väl genom att skapa en icke-linjär beslutsgräns.

k-Nearest Neighbors (k-NN) och Random Forest uppnådde något lägre resultat med testnoggrannheter på 94,7 % respektive 96,4 %. k-NN:s prestanda påverkades troligen av algoritmens känslighet för brus i datan och att varje prediktion kräver beräkning av avstånd till alla träningspunkter, vilket gör algoritmen mindre effektiv vid stora datamängder. Random Forest:s resultat visar på styrkan hos ensemblemetoder, men modellen riskerar att överanpassa när trädens djup och komplexitet ökar, trots användningen av begränsningar på djup och minsta antal prover för delning.

Den höga prestandan hos MLP kan kopplas till användningen av ReLU-aktivering och Adam-optimering, vilka hjälpte modellen att lära sig mönster snabbare och mer effektivt. Förbehandlingen av datan genom skalning och omvandling av bilder till vektorer var också en viktig faktor för att förbättra modellens inlärning. Resultaten visar att rätt val av hyperparametrar och en genomtänkt förbehandling av data är avgörande för att uppnå hög prestanda vid bildklassificering.

5 Slutsatser

Syftet med detta arbete var att jämföra prestandan hos fyra maskininlärningsmodeller – Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Random Forest och Multi-Layer Perceptron (MLP) – för bildklassificering av siffror med hjälp av MNIST-datasetet. Målet var att identifiera vilken modell som presterade bäst i termer av noggrannhet på validerings- och testdata samt att analysera hur förbehandling och val av hyperparametrar påverkade modellernas resultat.

Resultaten visar att MLP-modellen presterade bäst med en träffsäkerhet på 97,8 % på testdatan. Detta bekräftar att neurala nätverk, särskilt med ReLU-aktivering och Adam-optimering, har en stark förmåga att lära sig komplexa mönster i bilddata. SVM uppnådde också en hög träffsäkerhet på 96,6 %, vilket visar på modellens förmåga att skapa effektiva icke-linjära beslutsgränser. k-NN och Random Forest uppnådde något lägre resultat med träffsäkerheter på 94,7 % respektive 96,4 %, vilket visar att de är mindre effektiva än MLP och SVM för denna typ av bildklassificering.

Förbehandlingen av datan, inklusive skalning och omvandling av bilder till vektorer, var avgörande för att förbättra modellernas prestanda. Skalningen hjälpte framför allt MLP och SVM att konvergera snabbare och förbättra noggrannheten.

Sammantaget visar resultaten att MLP är den mest effektiva modellen för bildklassificering av siffror i MNIST-datasetet. Framtida arbete kan inkludera ytterligare optimering av hyperparametrar och utforskning av djupare nätverksstrukturer för att ytterligare förbättra prestandan. En annan

potentiell förbättring är att introducera dataaugmentation för att öka modellens generaliseringsförmåga till variationer i bilddata.

6 Teoretiska frågor

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

Träningsdata används för att modellen ska lära sig mönster i datan. Valideringsdata används för att finjustera modellen och undvika överanpassning. Testdata används för att utvärdera modellens prestanda på ny, osedd data.

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 ”validerings-dataset”?

Julia kan använda korsvalidering där datan delas upp i flera delar. Modellen tränas på vissa delar och testas på de andra för att identifiera vilken modell som presterar bäst.

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

Regressionsproblem innebär att förutsäga ett kontinuerligt värde, exempelvis bostadspriser eller temperaturer. Vanliga modeller är linjär regression, lasso och ridge regression.

4. Hur kan du tolka RMSE och vad används det till?

RMSE (Root Mean Squared Error) visar hur nära modellens förutsägelser är de verkliga värdena. Ett lägre RMSE innebär högre noggrannhet. Det används för att jämföra modeller vid regressions problem.

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

Klassificeringsproblem innebär att modellen förutsäger vilken kategori en observation tillhör. Vanliga modeller är SVM, k-NN, Random Forest och MLP. En confusion matrix visar antalet korrekta och felaktiga förutsägelser och används för att mäta prestanda.

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

K-means är en klusteringsalgoritm som grupperar data i k kluster. Algoritmen används bland annat för kundsegmentering och bildsegmentering.

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

Ordinal encoding = Tilldelar kategorier en ordning. Exempel: Liten (1), Mellan (2), Stor (3).

One-hot encoding = Omvandlar kategorier till binära värden. Exempel: Röd = [1, 0, 0], Grön = [0, 1, 0].

Dummy variable encoding = Liknar one-hot, men en kategori används som referens. Exempel: Röd = [1, 0], Grön = [0, 1] medan Blå = [0, 0].

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?

Julia har rätt. Nominaldata saknar ordning (exempelvis färger), medan ordinaldata har en naturlig ordning (exempelvis betyg).

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

Streamlit är ett ramverk för att bygga interaktiva appar i Python. Det används för att visualisera data och bygga dashboards för att presentera maskininlärningsresultat.

7 Självutvärdering

1. Utmaningar och hantering

En utmaning var att hitta rätt hyperparametrar, vilket tog tid eftersom olika inställningar påverkade modellernas prestanda på olika sätt. Jag hanterade det genom att testa flera parametrar och jämföra resultaten. En annan utmaning var att modellen presterade dåligt på bilder med inverterade färger. Jag försökte lösa det genom att justera förbehandlingen men valde till slut att acceptera begränsningen och nämna den i rapporten.

2. Betyg och motivering

Jag anser att jag förtjänar ett VG eftersom jag har implementerat flera modeller, jämfört deras prestanda noggrant och identifierat styrkor och svagheter. Jag har också byggt en fungerande Streamlit-app och analyserat resultaten på ett strukturerat sätt.

3. Feedback till Antonio

Det var en rolig men utmanande kurs.

Källförteckning

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.

Scikit-learn. (n.d.). Scikit-learn documentation. Hämtad från https://scikit-learn.org

Python Software Foundation. (n.d.). Python documentation. Hämtad från https://docs.python.org

Streamlit. (n.d.). Streamlit documentation. Hämtad från https://docs.streamlit.io