En Jämförande Analys av SVM och Random Forest Classifier på MNIST Databas

Parsan Amani

EC Utbildning

Kunskapskontroll 2 i Maskininlärning

2024–03

Abstract

Comparing the performance of two machine learning models, Support Vector Machine (SVM) and Random Forest Classifier, on the MNIST dataset, a vast collection of handwritten digits which has been used for training images processing systems. I apply the standard 70,000 image dataset, dividing it into training and testing sets to evaluate model accuracy and computational efficiency. My result reveal that while both models exhibit high levels of accuracy by helping GridSearch(for SVM in this case), the Random Forest Classifier slightly outperforms the SVM in terms of prediction accuracy. However, the SVM model demonstrates faster prediction times, making it more suitable for applications requiring real-time analysis. This research emphasizes the importance of model selection based on specific performance metrics and application requirements, offering insights into the strengths and limitations of each model within the context of handwritten digit recognition.

Förkortningar och Begrepp

RMSE: Root Mean Squared Error

SVM: Supported Vecktor Machine

RFC: Random Forest Classifier

Innehållsförteckning

Inledning

Inom området maskininlärning är klassificering av data en grundläggande uppgift. MNIST datasetet består av handskrivna siffror och har typiskt använt för att testa klassificeringsproblem.

Syftet med denna rapport är att jämföra Support Vecktor Machine (SVM)och Random Forest med avseende på deras effektivitet, noggranhet och tillämpbarhet på MNIST datasetet för att ge insikter om vilken metod som är mest lämplig för bildigenkänning av handskrift. För att uppfylla syftet så kommer följande frågeställning(ar) att besvaras:

1. Vilken av modellerna mellan SVM och RFC uppvisar högst noggrannhet med klassificering av handskrivna siffror i MNIST datasetet?
2. Hur påverkar modellernas parametrar deras prestanda i sifferigenkänning på MNIST datasetet?

Teori

En kort översikt genom MNIST-Dataset

MNIST-datasetet används framför allt för att träna olika algoritmer för bildigenkänning. Det är ett fundament inom maskininlärning Det erbjuder en utgångspunkt för att utveckla och testa algoritmer samt fungerar som en referenspunkt för att bedöma olika modellers prestanda. Detta är en utmärkt databas för personer som vill experimentera med att lära sig tekniker och metoder för mönsterigenkänning med verkliga data, samtidigt som de behöver lägga ned minimala ansträngningar på förbehandling och formatering. De ursprungliga svartvita bilderna från MNIST anpassades i storlek för att passa inom en 20x20 pixlars ruta, medan deras bildförhållande bevarades. Bilderna centrerades sedan inom en 28x28 pixelstor bildram genom att beräkna bildpunkternas masscentrum och justera bilden så att denna punkt hamnade i mitten av det 28x28 stora området.

1.2 Support Vecktor Machine

Support Vector Machine (SVM) är en effektiv modell för övervakad inlärning, huvudsakligen använd för klassificering men även för regression. Den identifierar det optimala hyperplanet som skiljer klasserna åt med största möjliga marginal. För att hantera icke-linjärt separerbara data använder SVM ett "kernel trick" för att projicera datan till ett högre dimensionellt rum där separation är möjlig. SVM stöder olika kärnfunktioner, såsom linjär, polynom, vilket gör den anpassningsbar för en rad olika tillämpningar från bildanalys till textklassificering. Trots att SVM presterar väl i högdimensionella utrymmen, kan dess komplexitet och behovet av att noggrant välja parametrar göra den mindre lämplig för mycket stora datamängder.

1.2.1 GridSearch för att finjustera hyperparameter

Det är en metod för att effektivt finjustera hyperparametrar genom att systematiskt testa olika kombinationer, vilket är kritiskt för att optimera modellers prestanda. I den här studien, har jag använt GridSearch att hitta och optimera min SVM-modellen. Genom att utforska ett brett urval av hyperparametervärden kan metoden automatiskt hitta de optimala inställningarna för en given datamängd, samtidigt som den minskar risken för modellens överanpassning med hjälp av korsvalidering.

1.3 Random Forest Classifier

Random Forest är en ensemblemetod för maskininlärning som använder en samling beslutsträd för att förbättra noggrannheten och minska risken för överanpassning. Varje träd tränas på slumpmässigt urvalda delmängder av datan och funktioner, vilket leder till en diversifierad modellensemble. Metoden aggregerar sedan trädens förutsägelser för att komma fram till ett mer stabilt och tillförlitligt slutresultat. Random Forest är känt för sin flexibilitet och kan användas för både klassificerings- och regressionsproblem.

Metod

Den här undersökningen tar sig an Python och Scikit-learn-biblioteket för att sätta upp och utvärdera både SVM (Support Vector Machine) och RFC (Random Forest Classifier) och använder MNIST-datasetet, “som består av 70 000 gråskalebilder av handskrivna siffror. Datasetet är uppdelat i 80% träningsdata och 20% testdata, vilket möjliggör omfattande träning av modellerna samt oberoende utvärdering (LeCun, n.d.)."

SVM-modellen optimerades genom Grid Search för att finjustera hyperparametrarna C-värde och gamma. För RFC, fokuserades optimeringen på antalet träd (n_estimators) och trädets maximala djup (max_depth).

Modellernas effektivitet bedömdes genom “Accuracy” på testsetet och beräkningstiden för träning. RMSE beräknades även som ett kompletterande mått för att bedöma modellernas prediktionsfel.

Denna metodik garanterar en balanserad och mångsidig utvärdering, som inte enbart baseras på modellernas förmåga att klassificera korrekt, utan också tar hänsyn till deras operationella effektivitet och precision i prediktioner.

Resultat och Diskussion

I denna studie jämfördes prestandan hos två välkända klassificeringsalgoritmer, Support Vector Machine (SVM) och Random Forest Classifier (RFC), på MNIST-datasetet för handskrivna siffror. SVM-modellen initierades med ett gamma-värde inställt på 'auto' och en fast random_state på 42. Denna konfiguration syftade till att ge SVM flexibiliteten att anpassa sig till datans komplexitet. Trots detta uppnådde SVM en lägre noggrannhet än förväntat, vilket antyder att den initiala inställningen kanske inte var optimal för MNIST-datasetets specifika egenskaper. Detta leder till en viktig insikt om betydelsen av hyperparameteroptimering, specifikt genom användning av Grid Search kombinerat med Cross-Validation, vilket senare visade sig vara avgörande för att finjustera SVM till en mer fördelaktig konfiguration.

Å andra sidan, när RFC implementerades med 100 estimatorer och samma random_state, visade den sig producera mer lovande resultat. Denna konfiguration, som bygger på en robust ensemble av beslutsträd, tycks ha gynnats av dess förmåga att effektivt hantera datasetets variation och komplexitet. Den inneboende mekanismen i Random Forest, som aggregerar ett stort antal trädets förutsägelser, bidrog till att minska risken för överanpassning – ett vanligt problem i maskininlärningsmodeller – och därmed öka dess allmänna noggrannhet.

Trots Random Forests starka prestation är det viktigt att inte enbart fokusera på noggrannhet som prestandamått. Vår initiala låga noggrannhet med SVM-modellen och den efterföljande förbättringen genom hyperparameteroptimering belyser den kritiska rollen av noggrant valda inställningar för att uppnå optimal modellprestanda. Dessutom understryker erfarenheten vikten av att överväga beräkningstid och modellkomplexitet, särskilt i realtidstillämpningar där dessa faktorer kan vara begränsande.

Slutsatserna från denna jämförelse understryker betydelsen av en balanserad bedömning som tar hänsyn till både noggrannhet och modellens praktiska genomförbarhet. Framtida forskning bör inte bara utforska andra algoritmer och hyperparameterinställningar utan också ta hänsyn till nya metoder för datanormalisering och förbehandling som kan förbättra modellernas förmåga att lära sig från MNIST-datasetet.

Genom att omfamna dessa insikter och fortsätta att utforska och optimera maskininlärningsmodellers inställningar, kan vi förbättra vår förmåga att automatisera och förbättra bildigenkänningstekniker,

vilket har breda tillämpningar från digitalisering av dokument till förbättring av användarinteraktioner med teknik.

RMSE för olika modeller	
SVM	1.22
RFC	0.77

Slutsatser

Vår jämförelse mellan Support Vector Machine (SVM) och Random Forest Classifier (RFC) på MNIST-datasetet visar att RFC generellt presterar bättre i noggrannhet för klassificering av handskrivna siffror. Denna överlägsenhet kan kopplas till RFC:s förmåga att hantera variation och komplexitet i datan effektivt, tack vare dess ensemblemetodik.

Vidare understryker studien vikten av hyperparameteroptimering. För SVM, bidrog användningen av Grid Search och Cross-Validation till att hitta optimala inställningar som förbättrade prestandan. Likaså påverkade antalet träd och deras maximala djup signifikant RFC:s prestanda, vilket visar hur kritiska dessa parametrar är för modellens noggrannhet.

Slutsatsen är att medan RFC i början erbjuder högre "Accuracy” för detta dataset, är en noggrann optimering av hyperparametrar avgörande för att maximera prestandan för både SVM och RFC.

Teoretiska frågor

1. Träning: Det är första steg att börja och träna datorer för att de kan lära sig regler för att hantera en uppgift utan att datorer har programmerats för detta.

   Validering: Validering är ett steg i utvecklingen av modeller. Det innebär att man kan bedöma hur väl en modell presterar på en separat datamängd som den inte tidigare har träffat för att se hur väl den kan tillämpas på nya, okända data.

   Test: Under detta steg utvärderas modellen på ett separat dataset, som kallas testdata från den original datan som vi har, och som modellen inte har haft tillgång till tidigare.

2. Julia kan använda Cross-Validation för att jämföra tre modellers prestanda (Performance) på träningsdata och välja den bästa optimering. Detta innebär att modellen tränas och utvärderas flera gånger med olika tränings- och valideringsuppsättningar för att undvika Overfitting.

3. Regression handlar om att förutse en kontinuerlig numerisk variabel baserat på "input features". Exempel på modeller är linjär regression, polynomisk regression och ridge regression. Tillämpningsområden inkluderar förutsägelse av huspriser, aktiekurser och försäljningsprognoser.

4. RMSE är ett mått på modellens prediktionsfel i förhållande till de verkliga värdena. Lägre RMSE-värden indikerar en bättre passning av modellen till data. Det används för att utvärdera modellens prestanda, jämföra olika modeller och identifiera områden där modellen kan förbättras.

5. Klassificeringsproblem innebär att modellen ska tilldela input till fördefinierade klasser. Exempel på modeller inkluderar logistisk regression och beslutsträd. Tillämpningsområden inkluderar medicinsk diagnos och e-postfiltrering.

   En Confusion Matrix är en tabell för att utvärdera modellens prestanda genom att visa antalet korrekta och felaktiga förutsägelser jämfört med de faktiska klasserna i testdatan. Det innebär att fyra delar; True Positive, True Negative, False Positive, False Negative.

6. K-means är en "Unsupervised" maskininlärningsalgoritm som används för att skapa grupper i data i olika "Cluster" baserat på deras likheter. Ett exempel på tillämpning är marknadssegmentering inom affärsanalys, där företag använder K-means för att kategorisera kunder baserat på deras beteende eller egenskaper, vilket möjliggör riktade marknadsföringsstrategier för att maximera försäljningen.

7. Ordinal encoding: Det används när kategoriska variabler har en ordning. Varje värde tilldelas en numerisk etikett enligt ordningen. Till exempel, vi har en kategori-variabel för utbildningsnivå med värdena "High School", "Bachelor's", "Master's", och "PhD". Med ordinal encoding kan vi tilldela dem numeriska värden som 0, 1, 2, och 3 respektive.

   One-hot encoding: Det används för kategoriska variabler utan inbördes ordning. Varje unikt värde omvandlas till en binär vektor där endast en dimension är 1 och resten är 0. Till Exempel, vi har en variabel för ögonfärg med värdena "blå", "grön" och "brun". En etta indikerar närvaron av den färgen för den specifika observationen.

   Dummy variable encoding: Det är en form av one-hot enenen encoding för kategoriska variabler med flera unika värden. Den skapar (n-1) binära variabler för n unika värden för att undvika problem med "multicollinearity".

8. Datan kan vara både "ordinal" och "nominal", beroende på kontexten och hur variablerna används. Så Julia har rätt. Till exempel, om Julia säger att "du är vackrast på festen om du har en röd skjorta", så tilldelar det en viss ordning till färgerna och gör dem därmed "ordinal".

9. Streamlit är ett programverktyg för att skapa webbsidor där man kan interagera med data och maskininlärningsprojekt. Det är gjort med Python, ett programmeringsspråk. Man kan använda Streamlit för att visa upp data på ett snyggt sätt, testa hur olika modeller för maskininlärning fungerar, och dela sina projekt med andra på internet. Det gör det enklare för folk att arbeta tillsammans med data och lära sig av varandra.

Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.
   Att jobba med koder och väntar för lång tid i början av skrivning koder och förstår hur kan man hantera dem modeller (särskilt SVM) var det en av de svåraste utmaningar, den “GridSearch” och “Cross-Validation" kod jag menar som skapar en pipeline som först skalar funktionerna med hjälp av StandardScaler och sedan tillämpar en SVM-klassificerare.

2. Vilket betyg du anser att du skall ha och varför.
   G och därför att har jag inte mycket tid att jobba med Streamlit applicationen.

3. Något du vill lyfta fram till Antonio?
   Jag vill tacka Antonio för att stötta oss elever i ett perfekt och praktiskt sätt. Den här boken som kursmaterial som vi använde var lite komplex att förstå men Antonio har lyckats att “highlight” viktigaste punkter på ett effektivt sätt.