Kontradiktoriska exempel inom djupinlärning

Edvard Söderback 37851
Kandidatavhandling inom datateknik
Handledare: Sepinoud Azimi
Fakulteten för naturvetenskaper och teknik
Åbo Akademi
2020

Innehåll

1. Inledning

Maskininlärning är en teknik som utvecklats väldigt mycket de senaste åren. Användningen av maskininlärning inom olika områden ökar också i en allt snabbare takt. Djupa neuronnät har visats kunna uppnå kompetens som är jämförbar med och i vissa fall till och med överträffar människor inom flera områden, exempelvis inom bildigenkänning. Före dessa neuronnät kan tas i bruk inom säkerhetskritiska områden krävs det dock att de inte kan vilseledas av en illvillig motpart. Det har visats att djupa neuronnät kan hantera slumpade störningar men inte störningar specifikt skapade för att vilseleda dem, även om dessa är helt osynliga för människor. Detta forskningsområde är därför ytterst relevant från ett säkerhetsperspektiv när det gäller praktisk tillämpning av djupa neuronnät. Utöver detta kan förståelse av dessa typer av störningar öka vår förståelse av hur djupa neuronnät fungerar.

Denna avhandling kommer huvudsakligen hantera vilseledande störningar inom bildigenkänning, så kallade kontradiktoriska exempel. I syfte av att ge en helhetsbild av området kommer både metoder att generera dessa och metoder att motverka dem gås igenom. För att ge läsaren en grund till detta kommer även grunder till maskininlärning tas upp.

2. Bakgrund

2.1 Maskininlärning

I takt med att datorer blir kraftfullare och billigare börjar de ges mer och mer uppgifter att lösa. För att en dator ska kunna lösa ett problem eller utföra en uppgift behöver den en serie med instruktioner för hur detta ska göras. Traditionella datorprogram är baserade på detta, man programmerar in explicita regler för hur en given mängd data ska hanteras för att producera resultatet man söker [1]. I vissa fall kan det dock vara näst intill omöjligt att beskriva hur ett problem ska lösas endast med hjälp av en mängd instruktioner. Hur känner man till exempel igen en person, tolkar tal eller skiljer siffror ifrån varandra?

En lösning på detta problem är då att låta datorn själv ta fram en algoritm för ett givet problem. Detta görs genom att ge datorn en stor mängd indata för vilken vi vet hur utdatan borde se ut och sedan låta maskinen själv lära sig vilka mönster och korrelationer som finns inom dessa data. Denna process, att låta en maskin lära sig lösa en uppgift utan att bli explicit programmerad att göra det, är vad som menas med maskininlärning [1]. En mer detaljerad beskrivning ges av Alpaydin i boken Introduction to Machine Learning:

> Maskininlärning är att programmera datorer för att optimera ett prestandamått med hjälp av exempeldata eller tidigare erfarenhet. Man har en modell definierad till vissa parametrar, och inlärningen är exekveringen av ett datorprogram för att optimera modellparametrarna utifrån träningsdata eller tidigare erfarenhet. Modellen kan vara förutsägande för att göra förutsägningar i framtiden, eller beskrivande för att få kunskap ur data, eller båda. [2, s.3] (min översättning)

Maskininlärning tillåter således användningen av datorer till uppgifter som inte kan beskrivas med vanliga algoritmer såsom datorseende och röstigenkänning. Även om maskininlärningsmodeller har visat mycket bra prestanda inom dessa områden lär de sällan sig

att basera sina beslut på samma mönster och korrelationer som människor gör. Mer om detta i kapitel 3.

2.1.1 Typer av maskininlärning

Olika typer av maskininlärning delas oftast in i tre olika kategorier enligt vilken typ av feedback som används, i övervakad, oövervakad och förstärkt inlärning.

Inom övervakad inlärning ges modellen en mängd indata med tillhörande utdata, för varje indata x finns utdata y som bestäms av en okänd funktion $y = f(x)$. Modellens uppgift är då att approximera denna funktion baserat på datamängden som ges till den [3]. Övervakad inlärning används huvudsakligen inom olika typer av datorseende såsom ansiktsigenkänning, bildklassificering eller handskriftstolkning.

Inom oövervakad inlärning ges modellen ingen förväntad output att optimera enligt utan ges endast en mängd indata. Modellens uppgift i detta fall är då att hitta olika mönster i datan, vanligen i form av grupperingar av indatan. Denna form av inlärning kan användas för att t.ex. gruppera kunder med liknande köpvanor i marknadsföringssyften. [4]

Inom förstärkt inlärning ges modellen inga explicita indata utan sätts istället i en miljö med ett specifikt mål att optimera enligt. Modellen bestämmer då själv vilka handlingar inom miljön som leder till positiva eller negativa resultat. Denna typ av inlärning används ofta för att lära en modell spela ett spel, exempelvis shack eller go. [4]

2.2 Djupinlärning

Även om maskininlärning idag överträffar människor inom en rad områden har detta självklart inte alltid varit fallet, tidiga maskininlärningsarkitekturer som beslutsträd eller stödvektormaskiner (eng. Support Vector Machine) är bristfälliga för mer komplexa uppgifter såsom ansiktsigenkänning eller röstigenkänning.

Djupa neuronnät (eng. Deep neural networks) är maskininlärningsarkitekturen som på senare tid har gjort det möjligt att lösa mera komplicerade problem. Denna maskininlärningsarkitektur är som namnet föreslår inspirerad av människans hjärna. Djupa neuronnät faller inom området djupinlärning som är ett delområde av maskininlärning. Djupinlärning är som Zhang et al. noterar i [5] en dåligt definierad term även om den har blivit väldigt populär det senaste årtiondet. En av definitionerna beskriver djupinlärning som

neuronnät med mer än två gömda lager. Patterson et al. påpekar i [1] att denna definition dock ger intrycket av att djupinlärning har existerat sedan 1980 fast det har krävts flera utvecklingar inom delområdet före man har uppnått det som idag förknippas med djupinlärning.

Djupa neuronnät är uppbyggda av tusentals ”neuroner” kallade noder som vidare är indelade i lager. Varje en av dessa noder tar in en mängd indata x som multipliceras med en tillhörande vikt w, denna viktade indata är därefter summerad för att få neuronens logit, $z = \sum_{i=1}^{n} w_i x_i$. En konstant b, kallad bias, adderas därefter till logiten och denna summa transformeras med en specifik nodfunktion för att producera nodens output. En nods output y kan då matematiskt definieras som följande: $y = f((\sum_{i=1}^{n} w_i x_i) + b)$. Denna output skickas senare vidare som input till nästa lagers noder och processen repeteras tills slutet av neuronnätet nås [6].

2.3 Optimering av djupinlärning

Innan ett neuronnät kan uppfylla en given funktion måste det först tränas med en stor mängd exempeldata, under träningsprocessen konfigureras alltså alla neuronvikter och biaser för att ge önskat resultat. Problemet med att konfigurera noder är något som snabbt växer i komplexitet när mängden noder växer. En stor mängd algoritmer och tekniker har därför utvecklats för att effektivt kunna lösa detta [6]. Några av dessa kommer att gås igenom närmare i resten av detta kapitel.

2.3.1 Förlustfunktioner

För att kunna konfigurera nodvikterna rätt krävs att man kan bedömma resultatet som neuronnätverket producerar på ett konstruktivt sätt. Detta görs med hjälp av olika förlustfunktioner, även kallade felfunktioner, som då är olika typer av mått för att bedömma hur långt ifrån önskat resultat outputten är. En felfunktion som vanligen används är medeltalet av felen i kvadrat. Denna kan formuleras som $E = \frac{1}{N}\sum_{i}^{N}(Y - Y')^2$ där Y är den korrekta outputten och Y' är nätverkets output [1].

2.3.2 Gradientnedstigning

Algoritmen som används för att hitta nodvikterna som minimerar förlustfunktionen kallas gradientnedstigning (eng. Gradient descent). Kortfattat funkar gradientnedstigningsalgoritmen på följande sätt. Först räknar man ut gradienten genom att derivera nätverkets förlustfunktion med de dåvarande nodvikterna. Därefter uppdaterar man nodvikterna enligt

riktningen på gradienten och upprepar processen tills man hittat ett minimivärde på förlustfunktionen [1]. Det finns ett flertal olika varianter av gradientnedstigning som bland annat skiljer sig i fråga om hur indata hanteras eller hur startvikterna väljs, men de följer alla denna grundprincip.

2.3.3 Bakåtpropagering

Att räkna ut gradienterna som används för att utföra gradientnedstigning är något som snabbt kan bli beräkningsmässigt tungt för stora neuronnätverk. För att kunna effektivt räkna ut gradienterna för enskilda noder används en algoritm som kallas bakåtpropagering. Som namnet antyder räknar man ut gradienterna genom att ha information att gå bakåt genom nätverket istället för framåt. Detta görs för att undvika att samma delterm inom ekvationen räknas ut flera gånger vilket vore beräkningsmässigt ineffektivt.

3. Skapande av kontradiktoriska exempel

3.1 Utforskande attacker

Utforskande attacker inom maskininlärning är ett forskningsområde som har växt explosionsartat sedan 2014 då Szegedy et al. [7] visade att djupa artificiella neuronnät kan tvingas felklassificera bilder med hjälp av minimala störningar i bilderna som är helt osynliga för människor. Szegedy et al. införde också uttrycket 'Adversarial example' för att beskriva dessa minimalt manipulerade bilder för att missleda artificella neuronnät. Denna term har i dagens läge blivit synonym med utforskande attacker mot djup maskininlärning, även inom andra områden än bildigenkänning. Utöver utforskande attacker existerar även förorsakande attacker som ämnar att producera fel inom maskininlärningsmodeller genom förändringar i träningsdatan. Dessa kommer dock inte gås igenom i denna avhandling.

3.1.1 Fullständig och ofullständig information

En stor del av forskningen på attacker av denna typ har gjorts med antagandet att motståndaren har tillgång till all information om neuronnätverket som attacken utförs på, det vill säga alla individuella nodvikter, träningsdata och arkitekturtypen som används i neuronnätet. Ett mer realistiskt antagande är att motståndaren endast har tillgång till klassificerarens utdata för given indata. Intuitivt kan man tänka sig att en motståndare med tillgång till all information om klassificeraren är ett mycket större hot än en med begränsad information, men det har dock visats att en motståndare med tillgång till minimal information är förvånantsvärt jämförbar med en motståndare med tillgång till all information. Mer om detta i stycke 3.3.6.

3.1.2 Riktade och oriktade attacker

Huruvida en attack försöker producera en viss typ av fel eller godtyckliga fel för en klassificerare är oftast orelevant när det gäller utforskande attacker. Detta beror på att oriktade attacker i de flesta fall endast är ett effektivare och mindre noggrant sätt att implementera riktade attacker [8]. Undantag till detta existerar där attack-algoritmen utnyttjar faktumet att den är oriktad istället för att maximera felfunktionen för en given input, exempelvis DeepFool [9].

3.2 Kvantifiering

För att producera ett kontradiktorisk exempel läggs en (av människor) oupptäckbar störning till en originalbild som ändrar dess klassificering. Huruvida denna störning kan upptäckas av människor eller inte är svårt att objektivt bedömma. Därför har man inom forskningen använt sig av olika L_p normer för att matematiskt bedöma hur nära en bild är en annan. Dessa normer används också inom regulariseringen av neuronnät för att justera felfunktioner, L_p normen för en vektor v beskrivs matematiskt som följande:

$$||v||_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{\frac{1}{p}}$$

De som huvudsakligen används inom området är L_0, L_2 och L_∞, de kan förklaras på följande sätt:

- L_0: Hur många icke-nollvärden en vektor har. För bilder är denna norm minimerad då ett så litet antal pixlar som möjligt ändras.
- L_2: Det euklidiska avståndet mellan vektorerna, det vill säga kvadratroten av summan av alla vektorvärdena i kvadrat. Inom bildigenkänning betyder detta att stora förändringar i pixelvärden har större påverkan än små ändringar men alla tas i beaktande.
- L_∞: Det största värdet av alla värden i en vektor. Denna tar då inte alls i beaktande antalet pixlar som ändras utan endast hur mycket de ändras.

Självfallet är detta inga perfekta mått på hur synlig eller utstickande en störning är för människor, exempelvis är oftast en stor förändring i en liten mängd pixlar lättare att upptäcka än en liten förändring i en stor mängd pixlar. Som en förbättring för dessa mått utvecklade Rozsa et al. [10] ett psykologiskt grundat mått som är baserat på hur människor upplever skillnader i bilder. Enligt skribentens uppfattning används detta mått dock sällan inom området.

Användbarheten för ett givet mått beror också till hög grad på situationen i vilken det används. Exempelvis är L_0 normen mer lämpad för områden där man intresserar sig av att bedömma individuella bitar, såsom nätverkskommunikation eller programvaruanalys.

3.3 Exempel på utforskande attacker

Härefter följer exempel på metoder för att generera utforskande attacker, dessa är valda delvis utifrån relevans till området och delvis utifrån hur illustrativa de är. Dessa metoder är anpassade för djupa maskininlärningsmodeller för bildigenkänning men kan även användas inom andra områden.

3.3.1 L-BFGS

Szegedy et al. [7] approximerade problemet med att producera kontradiktoriska exempel som följande:
Minimera $c \cdot ||x - x'||_2^2 + loss_f(x', t)$ så att $x' \in [0,1]^n$
där x' är den nya bilden som produceras genom att lägga till en störning till originalbilden x, t är output-klassen som man vill producera, c är en parameter som introduceras för att göra problemet lättare att hantera och $loss_f$ är klassificerarens förlustfunktion. De löser sedan problemet med hjälp av en optimeringsalgoritm kallad L-BFGS (Limited memory Broyden–Fletcher–Goldfarb–Shanno algorithm) som använder sig av gradienter. Ett optimalt värde för $c > 0$ hittas genom linjärsökning. Man löser alltså optimeringsproblemet flera gånger för olika värden på c och uppdaterar värdet därefter. På grund av att linjärsökning är beräkningsmässigt dyrt anses denna metod vara opraktisk [11].

3.3.2 FGSM

Goodfellow et al. [12] tog senare fram en metod som de döpte till ”fast gradient sign method” (FGSM). De menade att utforskande attacker mot djupa neuronnät existerade på grund av vissa neuronnätsmodellers linjäritet istället för deras icke-linjäritet som tidigare hypotiserats. Neuronnät som helheter är icke-linjära funktioner men Goodfellow et al. påpekar att många typer av neuronnät är designade så att de uppför sig linjärt eftersom detta förenklar optimeringsprocessen. Många små ändringar i indatan borde då leda till en stor ändring i utdatan, baserat på denna teori utvecklade de FGSM som är en väldigt snabb och enkel metod för att producera kontradiktoriska exempel. Denna metod kan beskrivas som följande:

$$x' = x - \varepsilon \cdot sign(\nabla loss_f(x,t))$$

där $\nabla loss_f(x,t)$ ger oss gradienten för modellens förlustfunktion och ε är en parameter som väljs sådant att störningen som produceras blir tillräckligt liten. I klarspråk så använder sig FGSM av förlustfunktionens gradient för att bestämma om ett pixelvärde ska ökas eller minskas för att modellen ska klassificera x som t, därefter adderas eller subtraheras ε till alla pixelvärden enligt detta. Detta problem kan effektivt räknas ut med bakåtpropagering. Eftersom denna algoritm väldigt lätt kan generera felklassificerade exempel visade Goodfellow et al. [12] att den kan användas som en effektiv regulariseringsmetod genom att konstant generera kontradiktoriska exempel som läggs till i träningsdatan.

3.3.3 BIM och PGD

I en studie om huruvida kontradiktoriska exempel fungerar som fysiska bilder vidareutvecklade Kurakin et al. [13] FGSM. Istället för att endast köra algoritmen en gång och producera en relativt stor störning så itereras algoritmen flera gånger med mindre justeringar för störningen. Denna metod kan beskrivas som följande:

$$x'_{i+1} = clip_{\varepsilon}\{x'_i + \alpha \cdot sign(\nabla loss_f(x_i, l))\}$$

där $clip_{\varepsilon}\{\}$ är en funktion som ser till att störningen hålls inom en viss gräns och α är storleken på störningarna som adderas eller subtraheras till pixelvärdena vid varje iteration. Skribenterna döpte denna metod till ”basic iterative method” (BIM). Märkbart är att fast den iterativa metoden visade sig överlägsen till FGSM inom den digitala domänen så fungerade sällan exempel genererade av BIM som fysiska bilder medan bilder genererade med FGSM bibehöll sin funktionalitet i de flesta fall.

Madry et al. [14] påpekade senare att denna metod är ekvivalent med projicerad gradientnedstigning (eng. projected gradient descent eller PGD) vilket är tidigare känd optimeringsmetod. Vidare menar Madry et al. [14] att PGD är en universell ”förstagrads motståndare”, d.v.s den starkaste attacken som använder sig av förstagrads information (gradienter) om modellen, denna hypotes stöds också av senare forskning [15].

3.3.4 DeepFool

Deepfool är en effektiv metod som togs fram av Moosavi-Dezfooli et al. [9] i syfte av att räkna ut hur robust en given klassificerare är mot utforskande attacker. Neuronnätet antas vara linjärt, därefter hittas den närmaste klassen till originalklassen, en störning läggs till originalbilden som skulle producera missklassificering ifall klassificeraren faktiskt var linjär, denna process repeteras sedan tills originalbilden felklassificeras. Eftersom algoritmen för Deepfool bygger på att kunna välja ut den klass som är närmast originalklassen

kan denna metod inte producera riktade attacker.

3.3.5 C&W

C&W attackerna är en serie av attacker som togs fram av Carlini och Wagner [8] för att motbevisa nätverksdestillation [16] som försvar mot utforskande attacker, dessa metoder är betydande eftersom de är de första attackerna inom djup maskininlärning som utvecklades för att kunna kringgå så många försvar som möjligt. Carlini och Wagners införde flera förbättringar jämfört med tidigare metoder.

Istället för att optimera indata enligt nätverkets förlustfunktion för den sökta etiketten utvärderades sex andra funktioner för detta syfte. Den effektivaste av dessa funktioner visades vara

$$f(x') = max(max\{Z(x')_i : i \neq t\} - Z(x')_t, -k) \tag{3.1}$$

där $Z(x)$ är nätverkets output före softmaxlagret och k är en variabel som införs för att kunna ställa in med vilken grad av säkerhet attacken felklassificeras. Funktionen jämför etiketten som nätverket ger störst värde för inputten med den sökta etiketten och returnerar positiva värden så länge den sökta etiketten inte ger störst värde.

Tidigare metoder använde sig av begränsningar på pixelvärdena för att se till att bilden som produceras är giltig. Dessa begränsningar begränsade även vilka optimeringsalgoritmer som kan användas för att lösa problemet. Genom att införa en ny variabel w och omformulera störningsvariabeln så att $r_i = \frac{1}{2}(\tanh(w_i)+1) - x_i$ ser man till att $x_i + r_i$ alltid har ett värde mellan 0 och 1 eftersom $\tanh(w_i)$ alltid har ett värde mellan -1 och 1. Detta problem kan nu lösas med effektivare optimeringsalgoritmer som annars inte kan hantera begränsade problem. Carlini och Wagner utvärderade tre olika sådana algoritmer: stokastisk gradientnedstigning (eng. stochastic gradient descent, SGD), gradientnedstigning med rörelsemängd och Adam [17]. De fann att alla tre producerade resultat av likadan kvalitet men Adam gjorde detta märkbart snabbare.

Deras metod kan då slutligen formaliseras som följande:

$$\min_w ||\frac{1}{2}(\tanh(w)+1) - x||_2^2 + c \cdot f(\frac{1}{2}(\tanh(w)+1))$$

där f är funktionen i ekvation 3.1 och c är en variabel som optimeras med binärsökning. Denna algoritm är anpassad för att minimera störningen under L_2 normen, Carlini och Wagner utvecklade även två andra attacker för L_0 och L_∞ normerna, dessa är mindre

relevanta inom området och kommer inte tas up i denna avhandling. Alla dessa attacker visades uppnå felklassificering i 100 % av fallen emot oförsvarade neuronnät, utöver detta så uppnåde de även samma grad av framgång mot destillerade nätverk där andra metoder endast lyckades skapa felklassificeringar för under 1 % av fallen. Vidare studier har också visat att C&W-attackerna är effektiva mot en stor mängd andra förslagna försvarsmetoder [18] [19].

3.3.6 Överförbarhet av kontradiktoriska exempel

Metoderna som presenterats hittills antar alla att motståndaren har tillgång till fullständig information om neuronnätet. I realistiska scenarion har motståndaren inte möjlighet att räkna ut gradienterna för en given input. Dessa metoder kan dock ändå användas för att effektivt generera kontradiktoriska exempel för en given modell även om motståndare har begränsad information om den. Detta beror på att kontradiktoriska exempel har visats ha förvånantsvärt hög grad av överförbarhet mellan olika maskininlärningsmodeller så länge dessa modeller har samma uppgift. Szegedy et al. [7] visade först att kontradiktoriska exempel skapade för en modell också fungerade mot andra modeller, både för modeller tränade på ett disjunkt dataset och för modeller tränade med andra hyperparametrar (antal nodlager, regulariseringmetoder, osv) än originalmodellen. Papernot et al. [20] undersökte senare detta fenomen närmare och fann följande:

- Överförbarheten håller i en stor del av fallen även om modellerna använder sig av olika maskininlärningstekniker.
- Genom att använda modellen som anfalles för att etikettera indata kan en motståndare enkelt skapa ett eget träningsdataset.
- Nollinformationsattacker mot alla typer av maskininlärningsmodeller är möjliga inom realistiska ramar.

För att bevisa dessa hypoteser utfördes attacker mot maskininlärningssystem från Google och Amazon. Med endast 800 förfrågningar till målmodellerna lyckades skribenterna producera attacker som felklassificerades i över 85% av fallen för båda modellerna utan någon information om modellerna i sig.

Denna tidiga forskning inom området fokuserade huvudsakligen på oriktade attacker. Liu et al. [21] fann senare att överföring av riktade attacker endast lyckas i under 2 % av fallen med dåvarande metoder. Denna studie fann också att överförbarhetsprincipen även håller för större dataset som ImageNet då tidigare studier endast använt sig av mindre dataset

som MINST och CIFAR. Liu et al. hypotiserade att om kontradiktoriska exempel felklassificeras på samma sätt av flera modeller så är det även sannolikt att ytterligare modeller felklassificerar det på samma sätt, vilket då skulle tillåta överföring av riktade attacker. Baserat på detta utvecklade de en algoritm för att producera riktade kontradiktoriska exempel som håller för en grupp med modeller, vilka motståndaren har fullständig tillgång till. Riktade attacker som utfördes med denna ensemble-baserade metod visades kunna överföras mellan olika djupa neuronnät i en majoritet av fallen, som förväntat fann man även att denna metod producerar starkare oriktade attacker.

3.3.7 ZOO

Även om attacker som utnyttjar överförbarheten är väldigt effektiva inom scenarion där motståndaren har begränsad information är detta inte det enda alternativet denne kan använda sig av i sådana situationer. Ett annat tillvägagångssätt är att approximera gradienterna med hjälp modellens svar på olika input. Chen et al. [22] var först med att utveckla en sådan attack, som de gav namnet ZOO (Zeroth order optimization (sv.nolltegradsoptimering)).

Chen et al. menar att det är naturligt att använda sig av nolltegrads-metoder för att attackera modeller med begränsad information, dock är dessa orimligt dyra att använda om direkt implementerade. Detta problem löses med en metod som liknar sättet man hanterar neuronnätsinlärning med stora dataset där man bryter ned optimeringen i mindre omgångar som tillåter effektivare beräkning. Med hjälp av detta samt några andra små ändringar i processen lyckades man ta fram en metod som är realistisk att använda. Under antagandet att en motståndare kan skicka en godtycklig mängd förfrågningar till neuronnätet som attackeras har ZOO visats prestera lika bra som vitalåde-attacker.

3.3.8 Enpixel-attack

De flesta studier som undersöker kontradiktoriska exempel använder sig av metoder som optimerar enligt normen L_2 eller L_∞. För att vidareutforska detta fenomen utvecklade Su et al. [23] en metod som optimerar för L_0 normen. Utöver detta skiljer sig denna attack även från andra attacker med att vara en av de första metoderna som genererar kontradiktoriska exempel utan användning av neuronnätets gradienter.

Attacken använder sig av en differentiellevolutionsalgoritm som optimerar för sannolikheten av den sökta klassificeringen. Kortfattat genererar algoritmen en uppsättning kandidatlösningar baserade på nuvarande lösningar. Kandidatlösningarna jämförs med lös-

ningen som de uppstod ifrån och den sämre lösningen förkastas. Denna process upprepas sedan tills en optimal lösning hittas. Metoden kräver alltså ingen information om maskininlärningsmodellen som anfalls utan endast sannolikhetensgraderna för modellens klassificeringar.

Attacken testades emot djupa neuronnäts klassificerare tränade på Kaggle, CIFAR-10 och ImageNet-dataset. I extremfallet där endast en pixel modifierades lyckades man inducera felklassificering för 67,97 % av bilderna från CIFAR-10 datasettet och 16.04% av ImageNet bilderna. Något som bör tas i beaktande med dessa resultat är att ImageNet-bilder har en upplösning som är 50 gånger större än CIFAR-10 bilder vilket då betyder att en enpixels-attack emot en ImageNet klassificerare är betydligt mer begränsad.

3.3.9 Gränsattack

Eftersom de flesta tidigare metoder använder sig av information som realistiskt kan undanhållas från en motståndare kan relevansen av utforskande attacker i verkligheten ifrågasättas. I syfte att närmare undersöka detta tog Brendel et al. [24] fram en attack som endast använder sig av den attackerade modellens slutgiltiga klassificering för en given input, vilket är mycket svårare att undanhålla en motståndare.

Idén bakom attackalgoritmen är i grunden okomplicerad. Man börjar med en bild som har klassificeringen man söker och därefter transformerar man stegvis denna bild till en bild med klassificering man vill efterlikna medan man ser till att klassificeringen inte ändras. Algoritmen håller sig alltså på beslutsgränsen mellan de två olika klassificeringarna och försöker hitta den punkt på gränsen där bilderna är mest lika varandra. Istället för att röra sig inifrån beslutsområdet ut till felklassificering börjar algoritmen utifrån och rör sig in mot den korrekta klassificeringen utan att gå över gränsen, vilket är olikt alla tidigare metoder.

Attackens simplicitet betyder också att den kan användas inom nästan vilka områden som helst. Det enda som krävs av en motståndare är att denne har tillgång till en instans av data med klassificeringen som man vill inducera, vilket Brendel et al. anser är trivialt.

Attacken visades kunna producera resultat som är i klass med C&W-attacken som anses vara den starkaste helinformationsattacken. Nackdelen med denna metod är dock att den kräver en stor mängd förfrågningar till modellen som anfalls.

4. Försvarsmetoder

Om maskininlärning ska kunna användas inom områden där systemet kan utsättas för olika typer av attacker krävs då att man implementerar någon typ av försvar som stoppar eller försvårar en motståndares attackmöjligheter. Speciellt inom säkerhetskritiska system som till exempel självkörande bilar är detta relevant. Försvar kan grovt klassificeras in i tre huvudkategorier enligt [25]: Gradientmaskering, robust optimering och detektion av kontradiktoriska exempel. Härefter följer exempel på olika försvar inom dessa kategorier. Dessa exempel har valts för att vara illustrativa och är inte nödvändigtvis de effektivaste inom kategorin.

4.1 Robust optimering

Det mest självklara sättet att försvara mot kontradiktoriska exempel är förstås att se till att modellen inte felklassificerar attacken. Robust optimering innebär som namnet antyder att man optimerar modellen för att bli robustare mot attacker, detta kan göras antingen genom att modifiera neruonnätet eller indata.

4.1.1 Träning med kontradiktoriska exempel

En av de första och mer intuitiva metoderna som föreslogs som försvar mot kontradiktoriska exempel är att inkludera dessa i nätverkets träningsdata och således låta modellen träna sig själv att stå emot attacker. Den första studien som undersökte inkluderingen av kontradiktoriska exempel i träningsdatan gjordes av Szegedy et al. [7], de undersökte dock inte huruvida denna metod fungerade som ett försvar men visade att denna metod regulariserade nätverket.

Goodfellow et al. [12] påvisade senare att träning med kontradiktoriska exempel även gjorde neuronnätverket mer robust mot attacker. Genom att använda sig av den mycket

effektiva FGSM för att producera kontradiktoriska exempel kan nätverket själv generera och sedan lära sig av dessa. Träning med kontradiktoriska exempel är dock inget komplett försvar och har flera nackdelar i praktiken. Eftersom försvaret inte är adaptivt och endast gör nätverket mera robust mot den typen av attack som tränas emot kan man inte garantera att försvaret fungerar mot en helt annan typ av attack. Utöver detta ökar denna metod både träningstiden och träningsdatan som krävs för att träna nätverket och funkar inte om nätverket inte är tillräckligt kraftigt. [26]

4.1.2 Slumpade förändringar i indata

Ett annat karakteristiskt tillvägagångssätt att försvara mot attacker inom bildigenkänning är att modifiera indata innan det ges till nätverket. Eftersom den kraftfullaste typen av förstagradsattacker, iterativa metoder, har visats vara ömtåliga gentemot transformationer av bilden följer det intuitivt att den bästa typen av försvar utnyttjar detta faktum.

Ett exempel på ett sådant försvar utvecklades av Xie et al. [27]. De visade att en stor mängd attacker kan stoppas genom att först slumpmässigt ändra storlek på bilden inom ett visst intervall och därefter lägga till en igen slumpmässigt vald mängd nollpixlar runt bilden. På grund av att transformationsparametrarna är slumpmässigt valda vid körningstid kan en motståndare inte få tillgång till dessa och således inte heller anpassa sin attack enligt dem. Detta är väldigt relevant när det gäller att försvara mot förstagradsmotståndare som annars har tillgång till fullkomlig information. Till skillnad från träning med kontradiktoriska exempel kräver denna metod ingen väsentlig mängd resurser att implementera eller köra eftersom den bygger på enkla transformationer av indata.

Detta försvar visades förbättra korrekt klassificering av t.ex. C&W-attacken från 0.3 % till 97.7 % på ett InceptionResNet-v2 nätverk. Ökningen av korrekt klassificering av den svagare FGSM-attacken visades dock endast vara från 61.2 % till 71.3 %. Xie et al. noterar att eftersom FGSM-attacken kan rimligt stoppas av kontradiktorisk träning borde en kombination av slumpmässig transformering och kontradiktorisk träning bilda ett effektivt försvar mot alla attacker. De visade att ett sådant försvar effektivt försvarade mot FGSM, DeepFool och C&W-attacken med en korrekt klassificering på minst 94.5 % av attackerna.

Även om detta försvar först verkar mycket lovande är det inget lösning på problemet med kontradiktoriska exempel. Det har tidigare visats att det är möjligt att producera kontradiktoriska exempel i form av 3D-objekt som felklassificeras från nästan alla perspektiv [28]. Dessa kontradiktoriska exempel kan således stå emot transformationer som är myc-

ket större än endast storleksändring. Från detta följer då att samma metod kan användas för att producera kontradiktoriska exempel som står emot detta försvar. Athalye et al. [29] visade senare att detta påstående stämmer. Genom att använda sig av samma metod som används för att producera kontradiktoriska 3D-objekt utformade de en attack som helt genomgår detta försvar, d.v.s felklassificering i 100 % av fallen.

4.1.3 Bevisbaserade försvar

Eftersom de flesta försvar inom området inte kan bevisa att de stoppar alla typer av attacker vore det oansvarigt att använda dessa inom säkerhetskritiska områden även om de kan uppnå goda resultat i de flesta fall. För att lösa detta problem har olika forskningsgrupper utvecklat försvar som bevisligen uppnår resultat inom en viss gräns för störningar under en viss gräns.

Gemensamt för alla typer av försvar inom denna kategori är att de kräver stora mängder beräkningar [30]. Mängden beräkningar som krävs ökar oftast även väldigt fort i takt med att nätverken blir större. Eftersom beräkningsmängden fort blir restriktiv så har de flesta studier inom denna kategori gjorts på mindre komplexa dataset som MINST och CIFAR.

Ett exempel på ett sådant försvar är en metod som togs fram av Raghunathan et al. [30]. Denna metod bygger på semidefinit approximering (eng. Semidefinit relaxation) och producerar ett certifikat för ett givet nätverk och input. Certifikatet intygar att det inte existerar någon typ av attack som kan inducera felklassificering i över en viss procent av fallen för en given störningsgrad. En intressant aspekt med detta certifikat är att det är deriverbart vilket betyder att det kan optimeras ihop med nätverksparametrarna för att skapa ett effektivare försvar. Ett annat certifikatbaserat försvar är PixelDP [31] som bygger på kryptografiskt inspirerade metoder. Utmärkande för PixelDP är att denna metod kan användas på större dataset som ImageNet.

4.2 Detektering av kontradiktoriska exempel

Eftersom korrekt klassificering av kontradiktoriska exempel har visats vara svårt att uppnå inom realistiska ramar använder vissa försvar ett mindre ambitiöst tillvägagångssätt. Att upptäcka kontradiktoriska exempel och därefter förkasta dem kan rimligtvis antagas vara en lättare uppgift än att korrekt klassificera dem. Det har dock visats att detta påstående inte nödvändigtvis stämmer [19]. Härefter följer exempel på olika försvarstyper inom denna kategori.

4.2.1 Neuronnätsdetektorer

Likt försvar som bygger på modifierad träning av neuronnät för att korrekt klassificera kontradiktoriska exempel har även försvar som använder sig av neuronnät för att identifiera kontradiktoriska exempel utvecklats. Gong et al. [32] tog fram en metod som använder sig av en binärklassificerare som skiljer på naturlig indata och kontradiktoriska exempel och förkastar de senare om sådana hittas. De fann att denna metod kunde stoppa svagare attacker såsom FGSM i 99 % av fallen, dock endast på MINST-datasettet vilket anses vara det lättaste att försvara. En variant av detta försvar som fungerar på CIFAR-datasettet utvecklades av Metzen et al. [33]. Istället för att använda bilden som klassificeras som input använder deras metod sig av data från neuronnätets alla gömda neuronlager, denna metod visades identifiera svagare attacker i över 80 % av fallen.

Carlini och Wagner [19] påvisade senare att dessa försvar inte står emot starkare attacker såsom C&W-attacken. De anser att användningen av neuronnät för att klassificera kontradiktoriska exempel är ett av de svagare tillvägagångssätten att producera försvar. De påpekar att om kontradiktoriska exempel kan lura en klassificerare är det också troligt att de kan lura både en detektor och en klassificerare, vilket de också bevisar.

4.2.2 Statistikbaserade försvar

Grosse et al. visade i [34] att kontradiktoriska exempel och naturlig indata inte har samma statistiska distribution och kan således särskiljas med hjälp av statistikbaserade tester. Baserat på detta faktum föreslår de en försvarsmetod som i teorin kan upptäcka alla kontradiktoriska exempel. De påpekar dock att detta inte är realistiskt i praktiken eftersom det i vissa fall kräver en oändlig mängd indata. Genom approximering av ett statistiskthypotestest kallat maximalmedelvärdesavvikelse (eng. maximum mean discrepancy (MMD)) menar de att kontradiktoriska exempel kan skiljas från naturligt förekommande indata. De lyfter också fram att en av nackdelarna med statistikbaserade tester är att dessa inte kan identifiera enskilda kontradiktoriska exempel utan behöver påträffa flera instanser av dem före en attack kan påvisas. För att motverka denna svaghet föreslår de att neuronnätet tränas om för att kunna upptäcka indata som faller utanför den naturliga distributionen av indata. Ett annat exempel på ett försvar som använder sig av distributionsskillnaden mellan kontradiktoriska exempel och naturlig indata togs fram av Feinman et al. [35] som istället för att jämföra skillnader i indata jämför skillnader i neuronnätets sista gömda nodlager. Carlini och Wagner [19] visade även här att dessa försvar inte står emot starkare och mera anpassningsbara attacker som C&W-attacken.

4.3 Maskering av gradienter

Ett tredje distinkt tillvägagångssätt att försvara emot kontradiktoriska exempel är att försvåra själva generationsprocessen. Detta kan göras genom att modifiera neuronnätet på ett sådant sätt att en motståndare inte kan använda sig av nätverkets gradienter för att producera kontradiktoriska exempel.

Papernot et al. [36] var först med att utveckla en sådan försvarsmetod. De använde sig av en variant av nätverksdestillering, en metod som tidigare använts för att överföra kunskap från ett större neuronnät till ett mindre neuronnät för att reducera nätverkskomplexiteten. Deras variant förminskar dock inte nätverksstorleken utan gör istället nätverkets gradienter försvinnande små. Eftersom de flesta helinformationsattacker använder sig av gradienter för att generera kontradiktoriska exempel stoppar denna försvarsmetod alla oanpassade attacker som använder sig av nätverkets gradienter.

Även här upprepades samma mönster som med försvarsmetoderna i tidigare kapitell. Papernot et al. påstod att denna försvarmetod stoppar generation av upp till 98 % av kontradiktoriska exempel, varefter Carlini och Wagner [37] visade att denna metod inte stoppar anpassade attacker. De visade också att destillering som försvar inte heller stoppade överförda attacker. Något som bör påpekas med försvar som bygger på gradientobfuskering är att dessa metoder endast fungerar gentemot attacker som använder sig av nätverkets gradienter. Detta betyder då att metoder som inte använder sig av fullkomlig information om nätverket som attackeras förbigår dessa försvar. Från detta följer att objektivt svagare metoder är mera effektiva emot denna försvartyp, vilket i teorin betyder att mera anpassade helinformationsattacker också kringgår försvar av denna typ. Detta påstående stöds av Athalye et al. [29] som menar att gradientbaserade försvar är oeffektiva och motbevisade de flesta gradientbaserade försvar inom området med hjälp av anpassade helinformationsattacker.

4.4 Problem med försvar

Som även framkommer i tidigare kapitel existerar det för tillfället ingen försvarmetod som i praktiken kan garantera att en djupinlärningsmodell är säker. Även om de flesta försvar kan stoppa en mängd attacker kan en motståndare med tillräcklig tid och information alltid kringgå dessa. Detta beror delvis på att kontradiktoriska exempel är ett nytt forskningsområde och en stor del av försvarsforskningen utvärderar inte sina försvar ordentligt [38]. Carlini et al. lyfter fram att de flesta försvar endast utvärderas mot antingen svaga

eller oanpassade attacker vilket inte ger en verklig bild av försvaret.

Ett annat problem med försvar i dagsläget är att både attacker och försvar inom området nästan uteslutande optimerar enligt endast en L_p norm. Schott et al. [39] påpekar att en motståndare inom realistiska scenarion inte behöver optimera enligt endast en L_p norm utan kan använda sig av en kombination av alla för att producera en starkare attack. De lyfter också fram att till följd av detta så är dagens försvar oftast endast effektiva mot attacker som optimerar enligt samma L_p norm som försvaret.

Utöver nuvarande brister i försvarsforskningen så kommer motståndare även alltid ha flera övertag i denna situation. Eftersom generering av kontradiktoriska exempel och träningen av en maskininlärningsmodell är väldigt liknande processer kan både försvararen och motståndaren använda sig av liknande verktyg. Skillnaden är att medan försvararen måste optimera för alla indata och utdata behöver attackeraren i många fall endast optimera för en instans av indata och utdata, vilket självfallet är en mycket lättare uppgift. Det är även möjligt för en motståndare att anpassa attackmetoden enligt försvaret medan det omvända fallet i princip är omöjligt.

4.5 Undanhållande av information

Som konstaterades i det tidigare kapitlet har attackeraren en stor fördel i teorin, iallafall om man antar att motståndaren har fullkomlig informationen om neuronnätet som attackeras. I praktiken är detta dock ett stort antagande och en motståndares möjligheter kan drastiskt förminskas om information undanhålles från denne. Inom realistiska scenarion kommer en motståndare inte ha möjlighet att räkna ut neuronnätets gradienter vilket utesluter nästan alla helinformationsattacker. Dock har det visats att gränsattacken [24] kan uppnå resultat som är jämförbara med starka helinformationsattacker givet att motståndaren kan göra en mycket stor mängd förfrågningar till neuronnätet. Från detta följer att mängden förfrågningar som kan göras till ett nätverk borde begränsas för att försvåra skapande av kontradiktoriska exempel mot neuronnätet. Begränsning av förfrågningsmängden gör det också svårare för en motståndare att träna upp ett eget surrogatnätverk som grund för överförbarhetsattacker [20].

Eftersom mängden information som en motståndare har tillgång till har en så stor inverkan på dennes möjligheter bör man noggrant överväga hur mycket information man ger åt användaren, speciellt inom säkerhetskritiska miljöer.

5. Slutsats

Djupinlärning är ett nytt och alltmer relevant område inom maskininlärning som också är ett nytt och alltmer relevant forskningsområde. För att man ska kunna använda sig av djupinlärning inom säkerhetskritiska områden krävs en djupare förståelse för hur denna teknik uppför sig. Ett fenomen som inte bara tillåter oss att bättre förstå djupinlärning men också lyfter fram säkerhetsbristerna inom den är kontradiktoriska exempel, indata som manipulerats på ett sådant sätt att en djupinlärningsklassificerare inte korrekt kan hantera den.

Denna avhandling ger en överblick över olika metoder för att generera samt försvara mot dessa kontradiktoriska exempel för att ge en helhetsbild över området. Att försvara emot kontradiktoriska exempel är fortfarande ett öppet problem och en ständig kapplöpning pågår inom området där en ny försvarstyp överträffar en ny attackmetod som senare överträffas av ännu ett nytt försvar osv. Av att döma från forskningen i dagens läge kommer denna kapplöpning inte ta slut inom den närmaste framtiden. Kontradiktoriska exempel kommer även bli ett mer och mer relevant fenomen i takt med att djupinlärning tas i bruk inom flera områden.

Litteratur

[1] Josh Patterson och Adam Gibson. *Deep learning: A practitioner's approach*. "O'Reilly Media, Inc.", 2017.

[2] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, 2014.

[3] Stuart Russell och Peter Norvig. *Artificial intelligence: a modern approach*. 2002.

[4] Yuxi Hayden Liu. *Python Machine Learning By Example*. Packt Publishing Ltd, 2017.

[5] W. J. Zhang m. fl. "On Definition of Deep Learning". I: *2018 World Automation Congress (WAC)*. 2018, s. 1–5.

[6] Nikhil Buduma och Nicholas Locascio. *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms*. Ö'Reilly Media, Inc.", 2017.

[7] Christian Szegedy m. fl. *Intriguing properties of neural networks*. 2013. arXiv: 1312.6199 [cs.CV].

[8] N. Carlini och D. Wagner. "Towards Evaluating the Robustness of Neural Networks". I: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, s. 39–57.

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi och Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks". I: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, s. 2574–2582.

[10] Andras Rozsa, Ethan M Rudd och Terrance E Boult. "Adversarial diversity and hard positive generation". I: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, s. 25–32.

[11] X. Yuan m. fl. "Adversarial Examples: Attacks and Defenses for Deep Learning". I: *IEEE Transactions on Neural Networks and Learning Systems* 30.9 (2019), s. 2805–2824.

[12] Ian J. Goodfellow, Jonathon Shlens och Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2014. arXiv: 1412.6572 [stat.ML].

[13] Alexey Kurakin, Ian Goodfellow och Samy Bengio. *Adversarial examples in the physical world*. 2016. arXiv: 1607.02533 [cs.CV].

[14] Aleksander Madry m. fl. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2017. arXiv: 1706.06083 [stat.ML].

[15] Nicholas Carlini m. fl. *Provably Minimally-Distorted Adversarial Examples*. 2017. arXiv: 1709.10207 [cs.LG].

[16] Nicolas Papernot m. fl. "Distillation as a defense to adversarial perturbations against deep neural networks". I: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, s. 582–597.

[17] Diederik P Kingma och Jimmy Ba. "Adam: A method for stochastic optimization". I: *arXiv preprint arXiv:1412.6980* (2014).

[18] Nicholas Carlini och David Wagner. *MagNet and "Efficient Defenses Against Adversarial Attacks" are Not Robust to Adversarial Examples*. 2017. arXiv: 1711.08478 [cs.LG].

[19] Nicholas Carlini och David Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods". I: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 2017, s. 3–14.

[20] Nicolas Papernot, Patrick McDaniel och Ian Goodfellow. *Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples*. 2016. arXiv: 1605.07277 [cs.CR].

[21] Yanpei Liu m. fl. *Delving into Transferable Adversarial Examples and Black-box Attacks*. 2016. arXiv: 1611.02770 [cs.LG].

[22] Pin-Yu Chen m. fl. "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models". I: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 2017, s. 15–26.

[23] Jiawei Su, Danilo Vasconcellos Vargas och Kouichi Sakurai. "One pixel attack for fooling deep neural networks". I: *IEEE Transactions on Evolutionary Computation* 23.5 (2019), s. 828–841.

[24] Wieland Brendel, Jonas Rauber och Matthias Bethge. *Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models*. 2017. arXiv: 1712.04248 [stat.ML].

[25] Han Xu m. fl. ”Adversarial attacks and defenses in images, graphs and text: A review”. I: *International Journal of Automation and Computing* 17.2 (2020), s. 151–178.

[26] N. Akhtar och A. Mian. ”Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey”. I: *IEEE Access* 6 (2018), s. 14410–14430. DOI: 10.1109/ACCESS.2018.2807385.

[27] Cihang Xie m. fl. ”Mitigating adversarial effects through randomization”. I: *arXiv preprint arXiv:1711.01991* (2017).

[28] Anish Athalye m. fl. ”Synthesizing robust adversarial examples”. I: *International conference on machine learning*. PMLR. 2018, s. 284–293.

[29] Anish Athalye, Nicholas Carlini och David Wagner. ”Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples”. I: *International Conference on Machine Learning*. PMLR. 2018, s. 274–283.

[30] Aditi Raghunathan, Jacob Steinhardt och Percy Liang. ”Certified defenses against adversarial examples”. I: *arXiv preprint arXiv:1801.09344* (2018).

[31] Mathias Lecuyer m. fl. ”Certified robustness to adversarial examples with differential privacy”. I: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2019, s. 656–672.

[32] Zhitao Gong, Wenlu Wang och Wei-Shinn Ku. ”Adversarial and clean data are not twins”. I: *arXiv preprint arXiv:1704.04960* (2017).

[33] Jan Hendrik Metzen m. fl. ”On detecting adversarial perturbations”. I: *arXiv preprint arXiv:1702.04267* (2017).

[34] Kathrin Grosse m. fl. ”On the (statistical) detection of adversarial examples”. I: *arXiv preprint arXiv:1702.06280* (2017).

[35] Reuben Feinman m. fl. ”Detecting adversarial samples from artifacts”. I: *arXiv preprint arXiv:1703.00410* (2017).

[36] N. Papernot m. fl. ”Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks”. I: *2016 IEEE Symposium on Security and Privacy (SP)*. 2016, s. 582–597. DOI: 10.1109/SP.2016.41.

[37] N. Carlini och D. Wagner. ”Towards Evaluating the Robustness of Neural Networks”. I: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, s. 39–57. DOI: 10.1109/SP.2017.49.

[38] Nicholas Carlini m. fl. ”On evaluating adversarial robustness”. I: *arXiv preprint arXiv:1902.06705* (2019).

[39] Lukas Schott m. fl. ”Towards the first adversarially robust neural network model on MNIST”. I: *arXiv preprint arXiv:1805.09190* (2018).