Omgekeerd rekenen is (g)een probleem.

Tristan van Leeuwen

Op 9 januari 2025 aanvaarde Tristan van Leeuwen zijn benoeming als hoogleraar *computationele inverse problemen* aan de Universiteit Utrecht. Zijn oratie is in verkorte vorm – en met meer technische details – hieronder opgenomen.

Seismologen gebruiken aardbevingsmetingen om te achterhalen waar een aardbeving precies plaatsvond en hoe sterk die was. In het ziekenhuis wordt uit een MRI-scan een 3D plaatje van de hersenen gemaakt. Astronomen maken met behulp van radiotelescoop een afbeelding van een zwart gat. Een rechercheur vergroot een opname van een beveiligingscamera om een nummerbord te kunnen lezen. Voordat er een nieuw windmolenpark wordt gebouwd, brengen ingenieurs de aardlagen onder de zeebodem in kaart. In al deze voorbeelden willen we iets terugrekenen uit metingen aan de hand van een wiskundig model. In het Engels noemen we dit een *Inverse Problem* wat vaak vertaald wordt als Inverse Probleem. Een betere vertaling is misschien *omgekeerd vraagstuk*. Omgekeerd rekenen is dus geen probleem, maar een vraagstuk. De essentie van een omgekeerd vraagstuk is dat we de oorzaak van een geobserveerd gevolg willen achterhalen. In zekere zin, is een omgekeerd vraagstuk het omgekeerde van een simulatie.

Toepassingen van, en onderzoek naar, omgekeerde vraagstukken hebben een lange geschiedenis. Een mooi voorbeeld is de ontdekking van de planeet Neptunus in 1846 [2]. Twee wetenschappers, Urbain Le Verrier en John Couch Adams, waren (onafhankelijk van elkaar) bezig een afwijking tussen de voorspelde en geobserveerde baan van een andere planeet, Uranus, te verklaren. Na lange berekeningen bleek een mogelijke verklaring hiervoor te liggen in het bestaan van een tot dan toe onbekende planeet. Deze berekeningen gaven ook precieze coördinaten voor deze onbekende planeet, en zo kon het bestaan ervan worden geverifieerd met een telescoop. Achteraf bleek dat vele wetenschappers al eerder of de afwijking hadden opgemerkt, of Neptunus hadden geobserveerd. De kracht van de ontdekking van Le Verrier en Adams lag dan ook in het samenbrengen van model en metingen, en het op een systematische manier zoeken naar een verklaring voor de discrepantie. En dat is de essentie van een omgekeerd vraagstuk.

Omgekeerde vraagstukken werden later ook vanuit een meer abstracte hoek belicht. Een mooi voorbeeld hiervan vind ik de vraag of men de vorm van een trommel kan horen. Deze vraag houdt wiskundigen al tientallen jaren bezig (zie bijvoorbeeld het voorwoord van [1] voor een beknopte geschiedenis en verdere referenties). Het blijkt dat veel praktische omgekeerde vraagstukken, van seis-

mologie tot materiaalkunde, kunnen worden herleid naar deze vraag. Met de opkomst van de computer in de jaren '50 van de vorige eeuw veranderde de aanpak voor het oplossen van omgekeerde vraagstukken. Waar wiskundigen eerder probeerde expliciete formules af te leiden om een omgekeerd vraagstuk op te lossen, werd het nu mogelijk om de computer met brute rekenkracht een oplossing te laten vinden. Dit werkt als volgt. Vanuit een beginschatting wordt het verwachte effect gesimuleerd, en vergeleken met de daadwerkelijke meting. Kloppen deze niet met elkaar? Dan wordt de beginschatting aangepast om zo tot een betere overeenkomst tussen simulatie en meting te komen. We herhalen deze stappen tot simulatie en meting met elkaar overeenkomen. In de afgelopen jaren zien we weer een kanteling en is er juist meer interesse voor het ontwikkelen van methoden die op meer directe manier een oplossing van het omgekeerde vraagstuk geven. Ook is er interesse in hoe de wiskundige inzichten over omgekeerde vraagstukken kunnen worden gebruikt om kunstmatige intelligentie te bestuderen.

Wat maakt omgekeerd rekenen zo uitdagend

We formuleren een omgekeerd vraagstuk in de praktijk vaak als een stelsel (niet-lineaire) vergelijkingen

$$\mathbf{y} = f(\mathbf{x}). \tag{1}$$

Hier staat $\mathbf{y} \in \mathbb{R}^m$ voor wat we meten, beschrijft $f : \mathbb{R}^n \to \mathbb{R}^m$ een verband tussen $\mathbf{x}$ en $\mathbf{y}$ dat we weten, en staat $\mathbf{x} \in \mathbb{R}^n$ natuurlijk voor de onbekende die we willen uitrekenen. Dit kan bijvoorbeeld een getal zijn, een coördinaat, of een plaatje. Het verband f is een wiskundig model. Soms is dit een expliciete formule, maar vaak is het een algoritme dat we kunnen uitvoeren voor gegeven $\mathbf{x}$ om de bijbehorende $\mathbf{y}$ uit te rekenen. 'Meten is weten' gaat voor omgekeerde vraagstukken dus niet op. Om $\mathbf{x}$ te weten uit de $\mathbf{y}$ die we meten, moeten we eerst het omgekeerde vraagstuk oplossen. Daarnaast moeten we er rekening mee houden dat de metingen vaak meetfouten bevatten, dus zelfs als we een $\mathbf{x}$ vinden zodat $f(\mathbf{x}) = \mathbf{y}$ is dat geen garantie dat dit het gewenste resultaat is.

De drie hoofdvragen van het omgekeerd rekenen die we ons nu stellen zijn

- Is er een $\mathbf{x}$ zodat $\mathbf{y} = f(\mathbf{x})$ (existentie),
- Is er precies één zo'n $\mathbf{x}$ of zijn er meerdere die tot dezelfde $\mathbf{y}$ leiden (uniciteit),
- Hangt het antwoord $\mathbf{x}$ continue af van $\mathbf{y}$ (stabiliteit).

Wanneer het omgekeerde vraagstuk aan deze drie eisen voldoet dan noemen we het *goed gesteld*. Anders noemen we het *slecht gesteld*. Een concreet voorbeeld is gegeven in figuur 1.

Wanneer het omgekeerde vraagstuk goed gesteld is, bestaat er een eenduidige omgekeerde relatie

$$\mathbf{x} = g(\mathbf{y}), \tag{2}$$

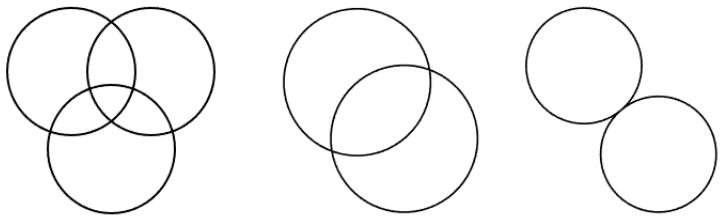

Figure 1: Voorbeelden van het omgekeerde vraagstuk in GPS-plaatsbepaling. De oplossing is het kruispunt van de cirkels. De situatie links heeft geen oplossing waar alle drie de cirkels kruisen. De situatie in het midden heeft twee oplossingen. Tot slot heeft de situatie rechts een instabiele oplossing, als we cirkels een klein beetje verschuiven hebben we twee oplossingen of geen oplossing.

waar g in zekere zin de inverse is van f. Het feit dat zo'n omgekeerde relatie bestaat wil nog niet zeggen dat we deze ook makkelijk kunnen uitrekenen. In de praktijk wordt g benaderd door middel van het herhaaldelijk toepassen van f. Het gevolg is dat het oplossen van een omgekeerd vraagstuk vaak veel meer rekentijd kost dan de bijbehorende simulatie. Een mooi voorbeeld hier van is het bepalen van de vierkantswortel van een getal. We lossen daarmee feitelijk de vergelijking $\mathbf{x}^2 = \mathbf{y}$ op om $\mathbf{y} = \sqrt{\mathbf{x}}$ te berekenen. In dit geval kunnen we $\mathbf{y}$ makkelijk uitrekenen voor gegeven $\mathbf{x}$ door een vermenigvuldiging uit te voeren. Om de vierkantswortel uit te rekenen, zullen in het algemeen echter een benadering moeten gebruiken. Een bekend algoritme om dit te doen is de recursie

$$\mathbf{x}_{k+1} = \tfrac{1}{2}\left(\mathbf{x}_k + \mathbf{y}/\mathbf{x}_k\right),$$

die door de Babyloniers ook al werd gebruikt. Dit soort iteratieve benaderingen worden nog steeds veelvuldig gebruikt om omgekeerde vraagstukken op te lossen.

Inverteren kun je leren

De vraag is nu of we iets kunnen doen aan deze asymmetrie tussen simulatie en inferentie (het oplossen van het omgekeerde vraagstuk). Een truc die we zelf gebruiken om bijvoorbeeld snel 2 getallen op elkaar te delen is dat we veel gevallen uit ons hoofd weten. Daarom leren we in groep 4 de tafels. Wanneer we een bepaalde berekening niet ons hoofd weten, proberen we deze terug te

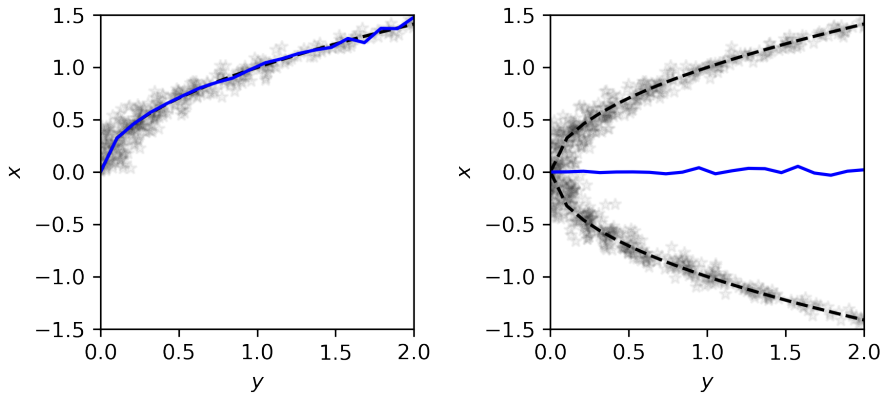

Figure 2: Voorbeeld van het leren van de vierkantswortel van een getal. Wanneer de training data alleen positieve getallen bevat (links) kunnen we de omgekeerde relatie (in blauw) goed benaderen. In het geval de training data zowel positieve als negatieve getallen bevat (rechts) dan is de omgekeerde relatie niet goed gedefinieerd en krijgen we een onbruikbaar resultaat.

brengen tot een som die we wel uit het hoofd weten. Kunnen we een computer niet ook leren om omgekeerde berekeningen snel te doen? Als we een groot aantal voorbeelden aan de computer laten zien kunnen we basis daarvan het omgekeerde verband $\mathbf{x} = g(\mathbf{y})$ leren door de punten met elkaar te verbinden. Dat is natuurlijk precies het principe waarop ook *machine learning* is gestoeld; we benaderen een relatie tussen in- en uitvoer met een neuraal netwerk op basis van een groot aantal voorbeelden. We beschouwen $\mathbf{x}$ en $\mathbf{y}$ nu als realisaties van kansvariabelen $\mathbf{X}, \mathbf{Y}$ met een kansverdeling $\pi_{\mathbf{X},\mathbf{Y}}$. In essentie wordt g geleerd door een optimalisatieprobleem op te lossen

$$\min_g \mathbb{E}_{\pi_{\mathbf{X},\mathbf{Y}}} \|g(\mathbf{Y}) - \mathbf{X}\|_2^2.$$

Dit levert een benadering op die feitelijk het conditionele gemiddelde berekend

$$g(\mathbf{y}) = \mathbb{E}_{\pi_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}} \mathbf{X}.$$

Wanneer er dus meerdere waarden van $\mathbf{x}$ tot (bijna) dezelfde $\mathbf{y}$ horen geeft deze benadering van g het gemiddelde van al deze $\mathbf{x}$-en. Een voorbeeld is gegeven in figuur 2. Deze direct aanpak kan dus in sommige gevallen een bruikbare benadering van de omgekeerde relatie opleveren. Echter moeten we erg oppassen

met het samenstellen van de training data en het toepassen van g op invoer die ver buiten de training data ligt.

Een Bayesiaanse blik

In de vorige sectie was het van belang de training data zorgvuldig samen te stellen om er zo zeker van te zijn dat er een eenduidige omgekeerde relatie uit kan worden geleerd. In veel toepassingen is dit echter niet wenselijk (het leidt mogelijk tot een vertekening in het antwoord) of zelfs niet mogelijk (omdat van tevoren niet duidelijk is welke voorbeelden moeten worden uitgesloten). Liever dan één antwoord weergeven (het conditionele gemiddelde) zouden we alle mogelijke $\mathbf{x}$ willen zien die bij een bepaalde $\mathbf{y}$ horen. Dit kan natuurlijk door de hele kansverdeling $\pi_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}$ als ons antwoord te beschouwen. In de Bayesiaanse statistiek noemen we deze kansverdeling vaak de *posterior*. De andere conditionele verdeling $\pi_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}$ wordt meestal de *likelihood* genoemd en geeft een stochastische beschrijving van de simulatie. De marginale verdeling $\pi_{\mathbf{X}}$ wordt in deze context vaak de *prior* genoemd en zegt iets over de selectie van $\mathbf{x}$-en die we waarschijnlijk achten. De Bayesiaanse blik biedt dus een mogelijkheid om de asymmetrie tussen simulatie en inferentie weg te werken; beide zijn immers gebaseerd op dezelfde onderliggende kansverdeling $\pi_{\mathbf{X},\mathbf{Y}}$. Daarnaast is het omgekeerde vraagstuk nu meestal goed gesteld; de benodigde posterior verdeling is in veel gevallen goed gedefinieerd, zelfs wanneer een eenduidige omgekeerde relatie niet bestaat [3].

Nu is dit wiskundig elegant, maar praktische gezien is een kansverdeling op zich niet zo bruikbaar. Uiteindelijk willen we namelijk realisaties hebben van de kansverdeling. Nu zijn er (al zo lang er computers zijn) algoritmen om realisaties te genereren uit een gegeven kansverdeling, waarvan de *Markov Chain Monte Carlo (MCMC)* methoden wellicht de bekendste zijn [4]. De uitdaging met het gebruik van deze methoden is dat de rekentijd vaak slecht schaalt met de dimensie (n, in het geval van de posterior).

Gelukkig biedt ook hier machine learning een uitkomst. Met name *generatieve modellen* geven ons de mogelijkheid om ingewikkelde kansverdelingen te leren uit voorbeelden, en om daar vervolgens nieuwe samples uit te generen. In deze context kunnen we een generatief model zien als een veralgemenisering van de omgekeerde relatie. In dit geval, geeft de omgekeerde relatie niet één deterministisch antwoord, maar een sample uit de conditionele verdeling $\pi_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}$:

$$\mathbf{x} = g(\boldsymbol{\xi}; \mathbf{y}), \quad \boldsymbol{\xi} \sim \pi_{\Xi},$$

waar $\boldsymbol{\xi} \in \mathbb{R}^n$ is verveeld volgens een zelf-te-kiezen kansverdeling π_{Ξ}, bijvoorbeeld een normale verdeling). We kunnen deze g in dat geval leren door het volgende optimalisatieprobleem op te lossen

$$\min_g \mathbb{E}_{\mathbf{y}\sim\pi_{\mathbf{Y}}} \mathrm{KL}\left(\pi_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}, g(\cdot;\mathbf{y})_{\#}\pi_{\Xi}\right),$$

waarbij $g(\cdot;\mathbf{y})_{\#}\pi_{\Xi}$ de *push-forward* is van de distributie π_{Ξ} door $g(\cdot;\mathbf{y})$ en $\mathrm{KL}(\cdot,\cdot)$ de Kullback-Leibler divergentie tussen twee kansverdelingen. Dit zorgt

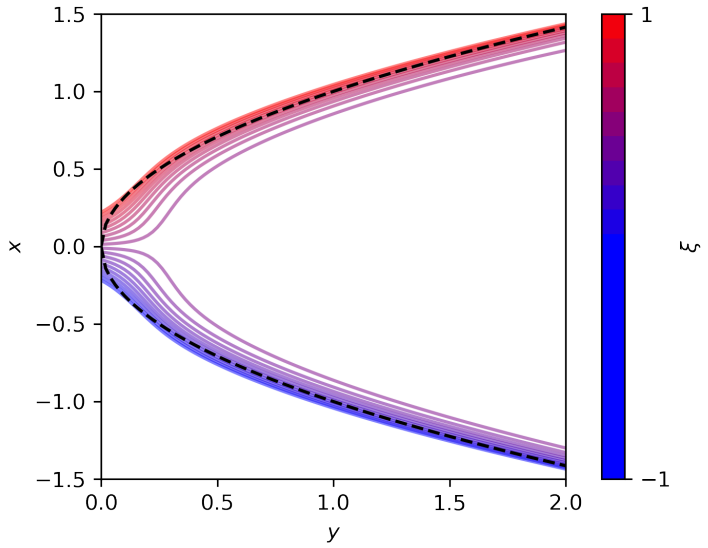

Figure 3: Voorbeeld van de geleerde omgekeerde relatie $g(\boldsymbol{\xi}, \mathbf{y})$ voor $a = 10$. We zien dat voor positieve en negatieve $\boldsymbol{\xi}$ de positieve resp. negatieve wortel wordt benaderd.

er in feite voor dat we een g vinden zodat de distributie van $\mathbf{X} = g(\Xi; \mathbf{y})$ de conditionele verdeling $\pi_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}$ benadert. Merk op dat dit vereist dat the relatie $\boldsymbol{\xi} \mapsto g(\boldsymbol{\xi}; \mathbf{y})$ inverteerbaar is, maar dat dit geen inverteerbare relatie tussen $\mathbf{x}$ en $\mathbf{y}$ veronderstelt. Op eenzelfde manier kunnen we ook een generatief model definiëren voor de simulatie

$$\mathbf{y} = f(\boldsymbol{\upsilon}; \mathbf{x}), \quad \boldsymbol{\upsilon} \sim \pi_{\Upsilon}.$$

Om een concreet voorbeeld te geven, definiëren we $\pi_{\mathbf{X},\mathbf{Y}}$ aan de hand van de volgende kansdichtsheidsfunctie

$$\rho_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi} \exp\left(-\tfrac{1}{2a}\left(\mathbf{y} - \mathbf{x}^2\right)^2 - \tfrac{1}{2}\mathbf{x}^2\right).$$

We kunnen het generatieve model voor de simulatie nu gemakkelijk opstellen:

$$f(\boldsymbol{\upsilon}; \mathbf{x}) = \mathbf{x}^2 + a\boldsymbol{\upsilon},$$

maar voor het generatieve model g vinden we niet zo'n elegante uitdrukking. We kunnen deze wel numeriek benaderen door het bovengenoemde optimalisatieprobleem op te lossen. Een voorbeeld is te zien in figuur 3.

Het wonderbaarlijke is, dat we deze twee generatieve modellen kunnen samenvoegen in één functie

$$(\boldsymbol{\xi}, \mathbf{y}) = s(\mathbf{x}, \boldsymbol{\upsilon}), \quad (\mathbf{x}, \boldsymbol{\upsilon}) = s^{-1}(\boldsymbol{\xi}, \mathbf{y}),$$

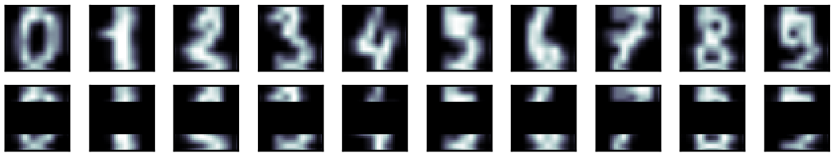

Figure 4: In dit voorbeeld willen we een handgeschreven getal lezen dat gedeeltelijk bedekt is. Hier zien we 10 voorbeelden van de training data; de volledige plaatjes $\mathbf{x}$ (boven) en de gedeeltelijk bedekte plaatjes y (onder). De volledige training data set definieert de kansverdeling $\pi_{\mathbf{X},\mathbf{Y}}$ waaruit we de afbeelding s leren.

waarbij s expliciet gegeven is in termen van f en g:

$$s(\mathbf{x}, \boldsymbol{\upsilon}) = \left(g^{-1}(\mathbf{x}; f(\boldsymbol{\upsilon}; \mathbf{x})), f(\boldsymbol{\upsilon}; \mathbf{x})\right),$$

met inverse

$$s^{-1}(\boldsymbol{\xi}, \mathbf{y}) = \left(g(\boldsymbol{\xi}; \mathbf{y}), f^{-1}(\mathbf{y}; g(\boldsymbol{\xi}; \mathbf{y}))\right).$$

In tegenstelling tot de deterministische formulering ($\mathbf{y} = f(\mathbf{x})$, $\mathbf{x} = g(\mathbf{y})$ met $g \neq f^{-1}$ in het algemeen), hebben in de Bayesiaanse formulering dus wel een inverteerbare relatie, s, tussen simulatie en inferentie.

Praktische implementatie

We kunnen nu op basis van voorbeelden $(\mathbf{x}, \mathbf{y}) \sim \pi_{\mathbf{X},\mathbf{Y}}$ de functie s leren, door een optimalisatieprobleem op te stellen die er voor zorgt dat de geleerde functie s de benodigde conditionele kansverdelingen benadert. Eenmaal geleerd, kunnen we s gebruiken voor simulaties en het oplossen van het omgekeerde vraagstuk. Een voorbeeld is te zien in figuren 4-7.

Het laatste puzzelstukje is om de functie s zo te parametriseren dat de inverse van s ook gemakkelijk uitgerekend kan worden uitgerekend. Dit lijkt misschien onmogelijk, maar aan de hand van een eenvoudig voorbeeld waarbij s een lineaire transformatie is kunnen we de denkrichting makkelijk illustreren. Op het eerste gezicht zouden we geneigd zijn s te parametriseren aan de hand van een volle matrix $S \in \mathbb{R}^{N \times N}$ met $N = n + m$. In dat geval is de complexiteit van de simulatie $\mathcal{O}(N^2)$ terwijl inferentie $\mathcal{O}(N^3)$ operaties kost. Als we transformatie echter parametriseren aan de hand van een LU-decompositie $S = LU$ waarbij L een benedendriehoeksmatrix en U bovendriehoeksmatrix is, dan is de complexiteit van simulatie en inferentie gelijk aan $\mathcal{O}(n^2)$. Een alternatieve manier is om S te parametriseren als $S = \exp(M)$ met inverse $S^{-1} = \exp(-M)$ en deze te benaderen als $\exp(M) \approx \left(I + k^{-1}M\right)^k$. De complexiteit van deze benadering is $\mathcal{O}(kN^2)$, voor zowel simulatie als inferentie. Beide aanpakken kunnen in zekere

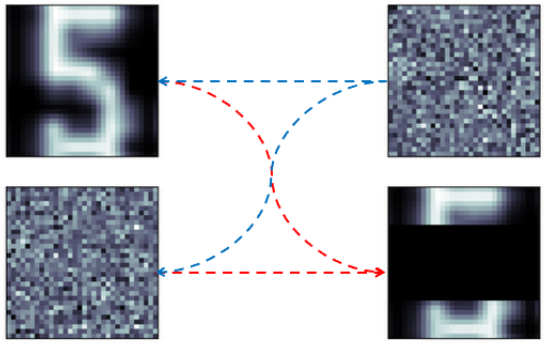

Figure 5: De afbeelding s combineert simulatie $\mathbf{y} = f(\mathbf{x}, \boldsymbol{\upsilon})$ (rood) en inferentie $\mathbf{x} = g(\boldsymbol{\xi}, \mathbf{y})$ (blauw) in één inverteerbare afbeelding $(\boldsymbol{\xi}, \mathbf{y}) = s(\mathbf{x}, \boldsymbol{\upsilon})$. In dit figuur is $\mathbf{x}$ een volledig plaatje (de 5 in dit geval) en $\mathbf{y}$ het gedeeltelijk afgedekte figuur. $\boldsymbol{\xi}$ en $\boldsymbol{\upsilon}$ bestaan uit normaal verdeelde ruis.

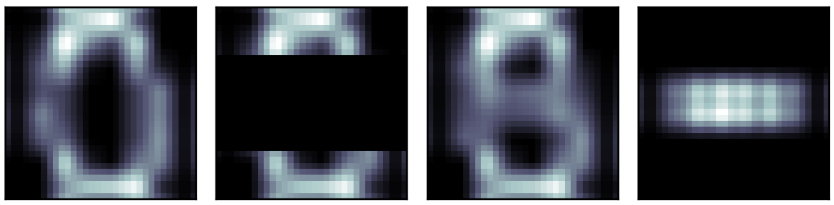

Figure 6: Voorbeeld van het generatieve model. Links zien we het oorspronkelijke plaatje, en daarnaast de bedekte versie die als invoer dient voor het generatieve model $g(\boldsymbol{\xi}; \mathbf{y})$. Het gemiddelde van een aantal realisaties is daarnaast te zien, en lijkt aan te geven dat het oorspronkelijke getal een 8 is. Als we echter naar de standaarddeviatie kijken zien we dat het middendeel erg onzeker is.

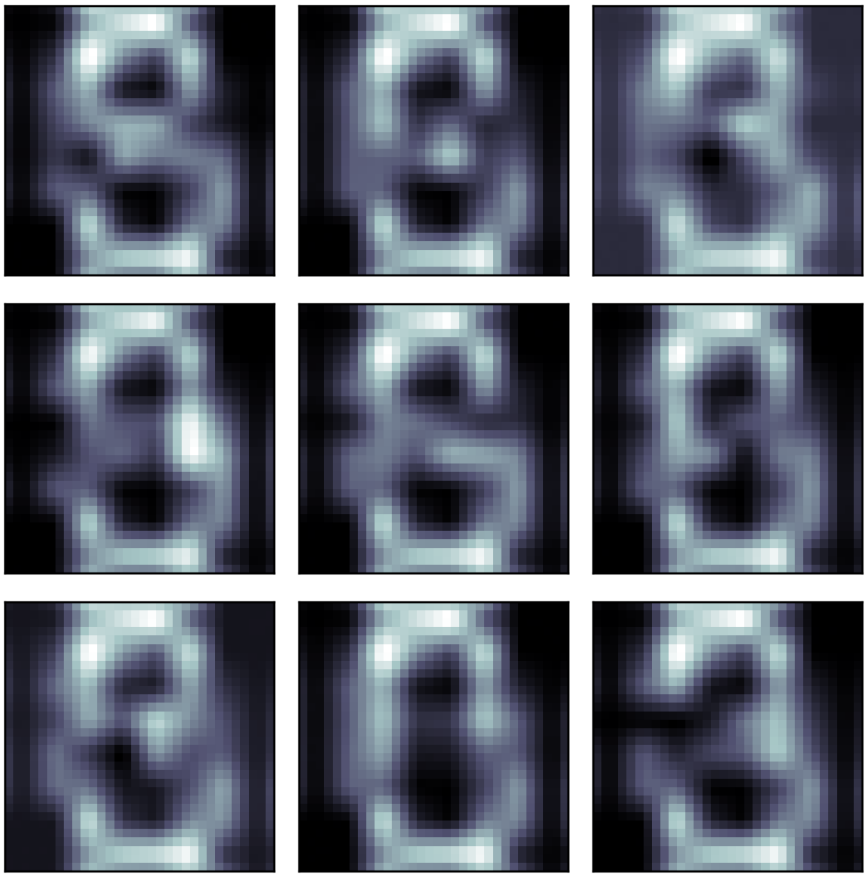

Figure 7: De kracht van het generatieve model wordt pas echt duidelijk wanneer we naar individuele samples kijken. We krijgen dan een beter beeld van de mogelijke scenario's: het getal kan bijvoorbeeld een 0, 3, of 8 zijn.

zin worden veralgemeniseerd naar niet-lineaire s door middel van speciale inverteerbare neurale netwerk architecturen of neurale differentiaalvergelijkingen [5].

De toekomst van omgekeerde vraagstukken

Dus hoe ziet omgekeerd rekenen er in de toekomst uit? Allereerst zie ik de scheidslijn tussen simulaties (reken $\mathbf{y}$ uit voor gegeven $\mathbf{x}$) en omgekeerd rekenen (reken $\mathbf{x}$ uit voor gegeven $\mathbf{y}$) vervagen. In beide gevallen willen we een verband leren uit voorbeelden. In beide gevallen willen we de onzekerheden die dit met zich meebrengt goed in kaart brengen. En in beide gevallen is het vaak van belang het antwoord snel en efficiënt uit te rekenen. Zo'n omkeerbare simulatie heeft een vooruit- en achteruitversnelling die met dezelfde efficiëntie het gevraagde antwoord in beide richtingen kan berekenen. Dit maakt het mogelijk dat nu alle beschikbare data bijdragen aan het oplossen van een specifiek omgekeerd vraagstuk waardoor een betrouwbaarder en representatiever resultaat wordt verkregen. Daarnaast zal het mogelijk worden om omgekeerde vraagstukken veel sneller op te lossen dan met de huidige methoden. Dit maakt het ook mogelijk om bijvoorbeeld geavanceerde driedimensionale beeldvorming in te zetten voor bagage-inspectie op Schiphol of kwaliteitsinspectie van appels aan de lopende band.

Zo'n omkeerbare simulatie lijkt dus een oplossing te zijn voor de twee grote uitdagingen (de lange rekentijd van stapsgewijs omgekeerd rekenen, en het bestaan van meerdere antwoorden). Wel moeten we ons zorgen maken over de benodigde rekenkracht en het bijbehorende energieverbruik voor het leren van zulke modellen. Maar hier liggen ook kansen. Eenmaal geleerd zou de omkeerbare simulatie immers best eens veel efficiënter kunnen zijn dan de huidige manier van rekenen. En het trainen kan, als er geen haast bij is, gebeuren op momenten dat er juist een overschot is aan groene stroom. Een ander aandachtspunt hier is hoe we communiceren over onzekerheden. Het is al tijdrovend voor bijvoorbeeld een radioloog om alle MRI-scans van een patiënt te bekijken zonder dat we alle mogelijke scenario's weergeven. We zullen de onzekerheden dus moeten samenvatten op een manier die voor de radioloog nuttig is. Ook hier zie ik een rol voor kunstmatige intelligentie; die kan het mogelijk maken dat de radioloog een dialoog aangaat met de berg aan data die nu beschikbaar is. Ook in andere toepassingen waar de interpretatie minder tijdsgebonden is, zoals bijvoorbeeld bij het aanleggen van een windmolenpark, zijn er uitdagingen omdat er aan geologische scenario's waarschijnlijkheden moeten worden toegekend. Aan de andere kant is informatie over de onzekerheden juist voor kunstmatige intelligentie van belang omdat deze niet zoals bijvoorbeeld een radioloog heeft geleerd om bepaalde aspecten van een scan als betrouwbaar of juist onbetrouwbaar te beoordelen. En uiteraard geldt dit ook buiten de medische toepassing. Overal waar kunstmatige intelligentie wordt gebruikt om data te interpreteren zou het goed zijn als deze op de hoogte is van de relatieve betrouwbaarheid van verschillende delen van de data. De verwachting is dat dit de resultaten veel

betrouwbaarder maakt.

References

[1] Gunther Cornelissen and Norbert Peyerimhoff. *Twisted Isospectrality, Homological Wideness, and Isometry: A Sample of Algebraic Methods in Isospectrality*. Springer Nature, 2023.

[2] Deborah Kent. The curious aftermath of neptune's discovery. *Physics Today*, 64(12):46–51, 12 2011.

[3] Jonas Latz. Bayesian inverse problems are usually well-posed. *SIAM Review*, 65(3):831–865, 2023.

[4] Christian Robert and George Casella. A short history of markov chain monte carlo: Subjective recollections from incomplete data. *Statistical Science*, 26(1):102, 2011.

[5] Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020.