COMPUTERPROGRAMMEURS KUNNEN NIET ALLE MOGELIJKE MENSELIJKE FOUTJES VOORZIEN; ZELFS EEN VERKEERDE MUISKLIK OF TOETSAANSLAG KAN SOMS VERREGAANDE GEVOLGEN HEBBEN. DAT SOORT ONTWERPFOUTEN EN VERGETELHEDEN WORDT VROEG OF LAAT WEL OPGESPOORD, MAAR VAAK ZIT HET SYSTEEM ZO COMPLEX IN ELKAAR DAT MOEILIJK TE ACHTERHALEN VALT IN WELKE STAP HET PRECIES FOUT IS GELOPEN. ARTIFICIEEL INTELLIGENTE SOFTWARE KAN EEN GROTE HULP ZIJN, MAAR SLAAGT ER NIET ALTIJD IN OM CAUSALE PATRONEN BLOOT TE LEGGEN EN WERKT SOMS ZELFS AVERECHTS. IN IEDER GEVAL VERGT OMGAAN MET NIEUWE TECHNOLOGIE EEN LEERPROCES DAT OOK STEEDS NIEUWE FOUTEN VOORTBRENGT. REGELMATIGE UPDATES ZIJN DUS ESSENTIEEL.



Een nieuwe update voor het systeem is beschikbaar

Adhemar Bultheel

Toetsenbord met update/synchroniseertoets. Foto: Bloomicon / Shutterstock

Er was een pauselijke bul nodig om in 1482 een afrondingsfout recht te zetten. Julius Caesar had als benadering voor de duur van een jaar 365,25 dagen van de Egyptenaren overgenomen, wat met een schrikkeljaar om de vier jaar een geheel aantal dagen per jaar toeliet. Maar in feite is een zonnejaar ongeveer 365,2421875 dagen. Als gevolg was ten tijde van paus Gregorius in 1582 het begin van de lente een dikke week opgeschoven, en daarom werden tien dagen geschrapt en voerde de paus de Gregoriaanse kalender in met nog steeds een schrikkeljaar om de 4 jaar behalve honderdvouden. Daardoor telde een jaar gemiddeld 365,2425 dagen. Maar dan nog loopt de klok 0,3 dagen voor na duizend jaar; daarom zijn veelvouden van 400 dan weer wel een schrikkeljaar.

Zo zie je maar dat een kleine afrondingsfout op termijn grote gevolgen kan hebben. Zolang er mensen achter het stuur zitten zullen de afwijkingen vroeg of laat wel rechtgezet worden, maar het wordt gevaarlijker als een computer blijft doortellen en afronden zonder nadenken. Een voorbeeld daarvan is de Vancouver Stock Exchange Index, ingevoerd in 1982. Die index werd zo'n 3000 keer per dag berekend, en daarbij werd het laatste cijfer van de berekening weggelaten. Daardoor werd telkens naar beneden afgerond, met als gevolg dat de index tegen eind 1983 was gezakt tot bijna de helft, terwijl er geen aanwijsbaar probleem op de markt was. Het duurt even voor dat soort fouten opgemerkt wordt omdat de kleine foutjes onder de alarmdrempel vallen. Dat is trouwens een principe waarmee internetcriminelen rijk proberen te worden: door 'verwaarloosbaar' kleine bedragen af te romen van heel veel bankrekeningen.

Wat beide voorbeelden leren is dat men correct moet afronden. Als men tot op een eenheid wil rekenen, dan zal men alles onder 0,5 moeten afronden naar beneden en alles erboven naar de volgende eenheid. De 0,5 zelf kan men bijvoorbeeld naar boven afronden als het vorige cijfer even is en naar beneden als het oneven is. Iets dergelijks geldt natuurlijk ook voor het laatste cijfer als men een groter aantal cijfers na de komma meeneemt. Het voorbeeld van de kalender toont ook aan dat dit er best zo veel mogelijk zijn. Het zal niet zo belangrijk zijn als je om de inhoud van je glas wijn te berekenen het getal π vervangt door 3,14, maar als dat niet nauwkeuriger gebeurt in de berekening van gps-coördinaten, dan kan jouw zelfrijdende auto weleens in het ravijn belanden.

Niemand kan echter exact π vangen met eindig veel cijfers, ook een computer niet. Onze digitale computers werken niet met decimale cijfers: in hun meedogenloze alles-of-nietsuniversum wordt alle informatie intern met bits, nullen en eentjes, voorgesteld. Alle informatie betekent zowel de instructies die uitgevoerd moeten worden als de gegevens

waarop die instructies toegepast worden en de geheugenadressen waarop de volgende instructie staat of waar de data te vinden zijn die verwerkt moeten worden. De bits worden gegroepeerd in woorden van 4, 8, 16, 32 of 64 bits, en verwerkt door processoren die in een paar klokcycli een instructie kunnen uitvoeren, waarbij de klok tegenwoordig tot 5 miljard cycli per seconde (5 GHz) haalt. Hier ligt een groot verschil met onze eerste twee voorbeelden. In het kalenderprobleem duurde het vele jaren alvorens men zich realiseerde dat er iets misliep en dan was er paus Gregorius die ingreep op het systeem. In het voorbeeld van de Stock Exchange zag men na een jaar dat er iets ontspoorde en kon men dat zonder al te veel schade corrigeren. Als men echter moderne computers hun gang laat gaan, en er gaat iets mis, dan gebeurt dat razendsnel. Hoewel deze systemen dikwijls het adjectief *smart* meekrijgen, kunnen ze onze 'little grey cells' (nog) niet echt evenaren, en beseffen ze niet altijd dat het de verkeerde kant opgaat. Extern ingrijpen voor het te laat is, is onmogelijk geworden omdat ze ons te snel af zijn.

In het meedogenloze alles-of-niets-universum van computers wordt alle informatie intern met nullen en eentjes voorgesteld

Niet alleen in snelheid, ook in geheugencapaciteit zijn computers in toenemende mate onze meerdere, wat ook moet omdat de informatie die we willen behandelen zo mogelijk nog sneller groeit. Maar ondanks dat 'giga'ntische geheugen kunnen alle (binaire) getallen die deze informatie voorstellen maar met eindig veel bits voorgesteld worden. En dat is nu precies een zwakte van de computer als die een relatief eenvoudige opdracht krijgt, zoals het tellen van een aantal verlopen seconden. In verschillende microprocessoren hield men de tijd bij door het aantal milliseconden te tellen. Dat stopte men in een register dat bijvoorbeeld 32 bits kon bevatten. Het grootste binaire getal dat men hierin kon bewaren bestond uit 32 eentjes achter elkaar gezet. Als dat milliseconden zijn, dan betekent dat in menselijke termen ongeveer 4,3 miljard of 49 dagen, 17 uur, 2 minuten en 47,296 seconden. Wanneer men het systeem uitschakelde, werd de teller telkens op nul gezet. Toen het eens niet gebeurde en het systeem voor 50 dagen liep, ontstond in 2004 complete chaos toen de Los Angeles Air Route Traffic Control alle radiocontact met de vliegtuigen verloor. Een dergelijk probleem zat in de software van de Boeing 787 Dreamliner, waar op gelijkaardige manier de tijd voor de generatoren voor de stroomvoorziening bijgehouden werd per 10 milliseconden in een 16-bitregister. Als dat 248 dagen continu bleef aanstaan zou het vliegtuig, ook midden in een vlucht, zonder stroom kunnen vallen. De 8-bittelregisters zijn trouwens ook de reden waarom een WhatsAppgroep niet meer dan 256 leden kan tellen en waarom men in Minecraft niet meer dan 256 blokken kan stapelen.

Ironisch genoeg doet een computer, bedoeld om rekenfouten van menselijke 'computers' te vermijden, niets anders dan fouten maken

Dit soort telproblemen kan men voorkomen door te werken met getallen die een veel groter bereik hebben. Dat zijn de getallen met bewegende komma. Men bewaart de grootteorde als een macht van 2. Die macht bewaart men bijvoorbeeld in 8 bits. De fijnere informatie, zeg maar de cijfers na de komma, stockeert men in de overige 24 bits van een 32-bitregister. Mochten het menselijke decimale cijfers zijn in plaats van bits, dan zou π bijvoorbeeld bewaard worden als 0,31415927(+1) waarbij de '0,' standaard voorop staat en dus niet bewaard moet worden en de '+1' op het einde '1 keer maal 10' betekent. Het verschil met de echte waarde van π is ongeveer 0,46410207(−7). De −7 betekent: voeg 7 nullen toe voor de eerste 4, of deel 7 keer door 10. Het verschil met π begint vanaf het achtste cijfer na de komma, maar ook dat verschil is weer afgerond op 8 cijfers: de laatste 7 in het kommagedeelte is een afronding. Bij elke bewerking wordt een foutje gemaakt; als men bijvoorbeeld getallen van 8 cijfers vermenigvuldigt bestaat het product uit 16 cijfers waarvan er maar 8 in het geheugen gestopt worden. En een computer maakt veel, heel veel, bewerkingen op zeer korte tijd, en elke bewerking draagt een foutje bij. Het is ironisch om vast te stellen dat een computer, bedoeld om rekenfouten van menselijke 'computers' te vermijden, eigenlijk niets anders doet dan fouten maken.

Nu hoor ik je al denken: wat kan die laatste bit nu een verschil maken? Wel, in een eenvoudige processor zoals gebruikt in een Patriot-raket wordt 0,1 (een tiende van een seconde) bewaard als een afgerond 24-bit-binair getal 0,00011001100110011001100, wat een fout is van ongeveer 0,000095 procent op het exacte oneindig repeterende getal. Dat is niet veel op een tiende van een seconde, maar tijdens de Golfoorlog stonden Patriots soms uren of dagen te wachten om een Scud te onderscheppen aan een snelheid van 6000 km/u. Met een fout van een derde van een seconde heeft men toen een Scud gemist met 500 meter, met vele doden als gevolg.

Naast deze inherente fouten op microschaal zijn er de programmeerfouten bij het softwareontwerp en natuurlijk ook de model- en ontwerpfouten van de ingenieurs. Zo is er de 'WalkieTalkie', een hoogbouw in Londen die met een gebogen glazen gevel het weerkaatste zonlicht concentreert en onderdelen van geparkeerde wagens doet smelten, of de 'Wobbly Bridge' (zoals Londenaars de Millennium Bridge noemen omdat die oorspronkelijk heen en weer slingerde toen er mensen over liepen). Maar dat soort fouten laten we buiten de softwarebeschouwingen van dit betoog. Wel belangrijk zijn 'ongelukjes' die men in de financiële wereld 'fat-finger errors' noemt, omdat een verkeerde toetsaanslag of muisklik een ongewoon grote transactie kan introduceren die in fracties van seconden wereldwijd computergestuurde reacties uitlokt die op hun beurt het systeem volkomen kunnen ontwrichten. Ook kan door oneigenlijk gebruik van een programma de computer in een toestand komen die door de programmeur niet voorzien is, zodat het systeem tilt slaat of mogelijk misbruikt wordt als het over gevoelige informatie gaat.

Daarom worden we constant geplaagd door updates van ons systeem die nodig zijn om dat soort vergetelheden en menselijke ontwerpfouten te corrigeren. Het is nu eenmaal zo dat, ondanks het uitgebreid testen vooraf, veel fouten slechts aan het licht komen nadat het een paar keer serieus is misgelopen. Met de toenemende complexiteit van de systemen die we gebruiken wordt het steeds moeilijker alle achterpoortjes te dichten. Het gebeurt ook vaak dat een bedrijf pas extra beveiliging inbouwt nadat het zelf, eventueel samen met een hele regio, door hackers gegijzeld werd. Erger wordt het als de computer zelf intelligente trekjes gaat krijgen en zelf gaat leren hoe dingen opgelost of beoordeeld moeten worden. Kunstmatige intelligentie is een alias voor de manier waarop de programmeur aan de machine uitlegt hoe die moet leren uit voorbeelden. Maar menselijke hersenen zijn het resultaat van een lang evolutionair proces, en zijn zo complex dat we de werking ervan nog niet helemaal doorgronden. Het leerproces in computers is bijvoorbeeld gebaseerd op een artificieel neuraal netwerk, een zeer elementaire babyversie van een heel klein onderdeeltje van onze hersenen dat met een zeer specifiek doel getraind wordt. Ook de evolutietheorie van Darwin wordt gebruikt om biologische of sociale systemen

te simuleren. Daarbij vertrekt men van het idee dat een initiële populatie van genen zich zal voortplanten volgens vooropgestelde selectiecriteria en wetmatigheden, die af en toe doorbroken kunnen worden door willekeurige mutaties. In beide gevallen steunt het resultaat op toevallige keuzes die in het leerproces gemaakt worden, en dan is het niet meer zo eenvoudig om, als het fout gaat, nog te achterhalen in welke stap het precies fout is gelopen. Trouwens, alleen al het begrip 'lukraak' is niet helemaal realiseerbaar voor een computer, want een willekeurig getal tussen nul en een is voor een mens een keuze uit oneindig veel getallen, terwijl een computer er maar eindig veel kan bewaren, waardoor de keuze beperkt is tot een eindig aantal mogelijkheden.

De gemiddelde persoon is een artificieel begrip

Dit stochastische, op toeval gebaseerde, gedrag is trouwens niet enkel een probleem voor de computer. Veel processen in de natuur, in de economie of in sociaal gedrag worden bepaald door zoveel parameters dat het doorgronden van de wetmatigheden enkel verkregen kan worden door een statistische analyse. De enorme hoeveelheid informatie die permanent gecreëerd wordt, en die door robots permanent afgeschuimd wordt, is niet meer door mensen beheersbaar en kan enkel door artificieel intelligente software verwerkt worden. Die kan leren om bepaalde kenmerken te identificeren, maar zal daar niet altijd de causaliteit van inzien of een waardeoordeel aan kunnen vastknopen.

Deze zogenaamd (artificieel) intelligente systemen worden ingezet bij rechtspraak, beurstransacties, advertenties, sollicitaties of de bestrijding van criminaliteit. Maar wat men wil voorkomen, zoals onrechtvaardigheid, bevooroordeeldheid of oninteressante reclame, werkt bij ondoordacht gebruik soms averechts en bevordert net wat men niet wil. Wetmatigheden die gelden voor grote groepen mag men niet toepassen op kleinere deelgroepen of individuen. De gemiddelde persoon is een artificieel begrip, want die persoon bestaat niet echt. Een werkschema kan opgesteld worden om het gebruik van tijd en plaats te optimaliseren, maar houdt er geen rekening mee dat arbeiders geen onvermoeibare automaten zijn die eenvoudige maar saaie bewerkingen eindeloos kunnen herhalen. Als de productiedoelstelling niet gehaald wordt, zal de werknemer ontslagen worden, maar als die een eigen succesvol concurrerend bedrijf opricht, zal de planningssoftware niet herzien worden. De beperking van dit systeem is dat de AI enkel en alleen gericht is op een optimale productie, terwijl de creativiteit van een werknemer, die een menselijke baas wellicht wel zou onderkennen, niet aan bod komt. Evenzeer kan een onderzoeker een slechte evaluatie krijgen op basis van een systeem dat enkel publicaties en impactfactoren telt, maar als die dan toch de Nobelprijs wint, is het nog maar de vraag of de evaluatiesoftware herschreven zal worden. Op basis van racistische gronden kunnen bepaalde bevolkingsgroepen in een rechtbank vaker als crimineel worden aangewezen. Om dat te voorkomen worden opgepakte misdadigers ondervraagd of ze al eerder zijn veroordeeld, of ze vrienden of buren hebben die al veroordeeld zijn, hun inkomen, hun opleidingsniveau, en dergelijke meer. Gegevens van verkeers- en bewakingscamera's worden eveneens meegenomen, waarbij men vergeet dat die laatste gegevens verzamelen over calamiteiten, en daarom talrijker aanwezig zijn in probleemwijken dan in andere stadsdelen. Als een zogenaamd onbevooroordeeld AI-systeem getraind wordt met deze gegevens en op basis van vorige uitspraken van de rechtbank, dan zal het net die racistische achtergrond leren en met een nefaste terugkoppeling met grotere waarschijnlijkheid een zwaardere straf voorstellen voor een voorbeeldige burger met de verkeerde achtergrond of die in de verkeerde buurt woont. Statistieken zijn eveneens bedrieglijk. De kans dat meer dan één kind van eenzelfde moeder aan wiegendood sterft is uiterst klein, en op basis daarvan kan een systeem voorstellen dat de moeder van moord beschuldigd wordt. Maar dat een kans zeer klein is, betekent nog niet dat het niet kan gebeuren.

We kunnen niet altijd weten of een fout 'op tijd' verbeterd werd

Wat zogenaamd objectief enkel op cijfers berust, zonder andere elementen in rekening te brengen, kan ook fout zijn. De systemen die zich baseren op medische gegevens en een verband leggen tussen het eten van ijsjes en gevallen van huidkanker (zie *Karakter* 71 voor een ander ijsjesvoorbeeld) zullen, zonder menselijke hulp, niet zo gauw het oorzakelijke verband leggen dat de ijsjeseters zonnekloppers zijn die zich zonder beschermende maatregelen veel ophouden in de felle zon. Monitoringsystemen voor de beurs gaan traditioneel uit van het feit dat mensen rationele wezens zijn, maar dat reacties op de beurs maar al te zeer emotioneel zijn is al dikwijls gebleken, en emoties zijn nu eenmaal niet erg logisch. Wat we leren uit dit overzicht is dat wij mensen heel slecht zijn in het leren uit onze fouten. Bovendien weten we niet altijd of een fout 'op tijd' verbeterd werd. Er is pas een publieke rapportering als er echt iets is misgelopen. In heel veel gevallen zullen commerciële bedrijven hun ingenieurs en werknemers om redenen van concurrentie een geheimhoudingsclausule doen ondertekenen waardoor fouten dikwijls stilzwijgend onder de mat geveegd kunnen worden. Wordt een fout ontdekt alvorens er een ongeluk gebeurt, dan wordt het systeem wel aangepast, maar dan zijn de gebruikers dikwijls vindingrijk genoeg om een extra vervelende controlestap te omzeilen, of vertrouwen ze zozeer op het systeem dat ze minder oplettend zijn en fouten maken die ze met meer verantwoordelijkheid niet zouden maken. Misschien worden systemen slimmer maar de gebruikers dommer. Verkeersborden vragen ons om het gps-systeem van de wagen uit te schakelen bij een gewijzigde verkeerssituatie, anders merken we de omleiding misschien niet op. Eventueel zorgt het systeem zelf omwille van een a posteriori aangebrachte correctie voor nieuwe fouten. Wereldwijde bedrijven die onze computers en mobiele multimediatoestellen beheren, hebben ons allen als onbezoldigde werknemers in dienst om fouten in hun systemen op te sporen. En we zijn nog niet aan het einde van deze evolutie.

Laat ik tot slot nog even terugkeren naar de computerhardware. Die is tegenwoordig zo geminiaturiseerd dat men met kwantumfysische effecten rekening moet houden. Dus onze traditionele opvatting dat computers deterministische machines zijn, waarop weliswaar stochastische processen gesimuleerd worden, moet herzien worden omdat computers zelf inherent stochastisch gedrag zouden kunnen vertonen. Dit idee wordt ten volle benut in kwantumcomputers, die als de veelbelovende en revolutionaire rekentechnologie van de toekomst voorgesteld worden. Maar omgaan met een nieuwe technologie vergt een leerproces en is gegarandeerd een bron van nieuwe fouten; je mag gerust veronderstellen dat er nog wel eens een kwantumberekende brug zal instorten of een kwantumgestuurde auto een kanaal voor een snelweg zal houden alvorens men een kleine maar noodzakelijke correctie zal aanbrengen. Een regelmatige update van je systeem zal nog steeds aanbevolen zijn.

Cathy O'Neil, *Weapons of Math Destruction. How Big Data Increases Inequality and Threatens Democracy*. (New York: Crown, 2016).
Matt Parker, *Humble Pi. A Comedy of Maths Errors*. (Londen: Allen Lane, 2019).

ADHEMAR BULTHEEL is emeritus professor van de KU Leuven. Hij doceerde gedurende vele jaren onder andere Inleiding tot de Numerieke Wiskunde voor beginnende studenten burgerlijk ingenieur.