

Testen met zelflerende en zelf-explorerende testtools

Door Rik Marselis en Sander Mol, maart 2017

Op welke manier gaat machine-intelligentie ons als testers in de toekomst verder helpen? Toen wij hier een tijdje over nadachten, kwamen we op het concept van zelflerende en zelf-explorerende testtools. Een grote stap voorbij de uitvoerende testtools van vandaag! Hieronder zetten wij deze gedachten uiteen. Aan het eind roepen we je op dit verder met ons te verkennen!

Testtools van nu, vooral voor het (vaak saaie) uitvoeringswerk

Testautomatisering automatiseert meestal de testcases die vooraf handmatig zijn opgesteld door de tester(s). Deze testgevallen hebben vooraf gedefinieerde resultaten. De tool verifieert of de werkelijke resultaten overeenkomen met de vooraf gedefinieerde resultaten. De testcases die worden geselecteerd, hebben als doel om een bepaalde dekking van alle mogelijkheden te behalen. Het aanpassen van het testobject betekent ook het aanpassen van de testcases in de tool.

De testwereld staat nu op het punt om nieuwe, andersoortige automatische testtools toe te gaan passen. Tools die niet vooraf instructie hoeven te krijgen, maar zelfstandig een informatiesysteem verkennen.

De zelflerende en zelf-explorerende tools van vandaag

De mogelijkheden om machines zelf te laten leren zijn de laatste jaren erg sterk toegenomen. Zo keek de wereld geschokt toe hoe het zelflerende AlphaGo de Koreaanse wereldkampioen van het spel Go overtuigend versloeg, 10 jaar eerder dan verwacht. En ook met poker heeft een computer al veel geld gewonnen van top-spelers. De makers van AlphaGo, Google's "Deepmind", heeft ook een ander programma dat uitblinkt in zelfleren en zelf-exploreren. Dit programma slaagt er zonder enige uitleg in om klassieke Atari spellen te onderzoeken, te leren en vervolgens tot in perfectie te spelen. Als voorbeeld kiezen we het spel 'breakout' waarvan je hieronder een screenshot ziet.



Het Deepmind-programma start het spel en heeft in het begin geen idee wat er gebeurt. Het concept van een bewegend balletje dat moet worden teruggedraaid is onbekend. Dat geldt ook voor het idee dat de rode pixels onder in beeld samen een object vormen dat te besturen is, en dat de punten gescoord worden door de gekleurde blokken weg te spelen. Alles is nieuw, dus alles moet verkend en geleerd worden.

Dit verkennen en leren gebeurt door het spel keer op keer te spelen. Na 10 minuten spelen krijgt het programma al enige handigheid in het besturen van de paddle. Na 300 spellen kan het programma net zo goed spelen als een menselijke speler dat zou kunnen. En na 500 spellen heeft het programma de optimale tactiek gevonden om het spel te winnen, door via de zijkant een opening te maken en de bal boven de hoogste rij blokken te laten stuiteren.

Met precies dezelfde tool kunnen ook andere spellen worden geleerd. Op Youtube vind je voorbeelden van Pacman en Super Mario. Zonder uitleg. Met één en dezelfde tool.

Zelflerende en zelf-explorerende testtools nemen ook het denkwerk over

Stel nu dat we deze Deepmind-tool geen Atari spel zouden voorschotelen, maar een object dat we willen testen, zoals een website. Wat zou zo'n tool kunnen doen? Wat zouden we ermee kunnen bereiken?

Laten we ons nu vooral richten op wat op korte termijn toepasbaar is: invoercontroles, opmaakcontroles, foutafhandeling en hier en daar de samenhang van waarden in twee velden. Ofwel de syntactische en semantische testen.

De tool zou zonder enige voorkennis gaan klikken, gaan typen, gaan slepen enzovoort. Het zou invoervelden ontdekken en daar allerlei soorten waarden gaan invoeren. Het zou willekeurige combinaties gaan maken van handelingen en waarden en zou ervaren wat dit voor resultaat oplevert. Het zou trends en patronen ontdekken en daarmee bepalen wat standaard gedrag is. En zodra de tool weet wat de standaard is, dan weet het ook wat afwijkend gedrag is. En dan kan de tool een rapport opleveren met een lijst van deze afwijkingen.

Dat is het punt waar wij als menselijke tester voorlopig nog nodig zullen zijn: het interpreteren van de afwijkingen en het onderscheid maken tussen gewenste afwijkingen en daadwerkelijke fouten. Uiteraard zal het zelflerende programma de volgende keer rekening houden met deze menselijke inbreng. Het zal de volgende testronde met een veel kortere lijst van door ons te interpreteren afwijkingen komen.

Net als bij de huidige testautomatisering, zit de kracht van dit concept in het herhalen van testen. In de volgende oplevering van het testobject zal de tool herkennen wat de veranderingen zijn en maakt opnieuw geen onderscheid tussen goed en fout. De menselijke tester kan dus zowel de gewenste veranderingen als eventuele regressiemeldingen beoordelen.

Trouwens, wie zegt dat dit een professionele tester moet zijn? We kunnen ook andere betrokkenen zoals de klant laten beoordelen, met steun en advies van de kunstmatige intelligentie.

Kunnen we de tool nog ergens mee helpen?

Hierboven ging het nog om de syntactische en semantische testen, maar de stap naar het automatisch testen van logische verwerkingen en procesflows is heel goed denkbaar. Zeker als we de tool een beetje op weg helpen door verschillende 'inputs' mee te geven. In principe zou de tool, na duizenden uren exploreren en leren, (bijna) alles zelf kunnen bedenken. De belangrijkste reden om inputs mee te geven is dus efficiëntie.

- We zouden testdata mee kunnen geven, zoals correcte postcodes en bestaande adressen, het formaat van telefoonnummers en bestaande gebruikersnamen en wachtwoorden.
- We kunnen requirements en ontwerpen aanleveren, het liefst in een gestructureerde vorm. De tool zal dan proberen te herkennen welke invoer- en uitvoerwaarden er in deze testbasis staan, en proberen dit te valideren in het testobject.

- We zouden, na een eerste testronde, kunnen aangeven welke van de ontdekte functionaliteiten we het belangrijkste vinden, zodat de tool zich daar de volgende testronde meer op richt binnen de gegeven tijd.

Daarnaast is er nog een hulpmiddel dat wij als professionele testers graag willen toevoegen: testontwerptechnieken. Hiermee kunnen we sturen op een bepaalde dekking. Dit soort technieken schieten er nu in de praktijk vaak bij in, dus het zou veel helpen als een tool hierin kan faciliteren. Bovendien: doordat de tool in principe een oneindig aantal testen kan doen, kan AI ook zelf tot nieuwe technieken komen, misschien zelfs beter dan wij als mensen kunnen bedenken. Denk maar aan de hierboven beschreven truc die de tool zelf bedacht om het spel Breakout het meest efficiënt te spelen. De beperking is op dit moment vooral de tijd die de tool en het testobject nodig hebben om de vele mogelijkheden uit te proberen.



Kan de tool zichzelf verder helpen?

Naast de menselijke 'inputs' is er nog een héél interessante 'input': de eerdere testen van de tool zelf. Op die manier kan de tool véél efficiënter testen tijdens een volgende testronde. Vergelijk het met het 500ste spelletje Breakout. De ervaring van de tool hoeft zich natuurlijk niet te richten op één testobject. Als we bijvoorbeeld een website hebben getest en we hebben als mens aangegeven welk soort afwijkingen we wel of niet wenselijk vinden, dan zal de tool bij de tweede website deze ervaring meenemen. Denk hierbij aan voorbeelden als gebruiksvriendelijkheid en toegankelijkheid voor bijvoorbeeld kleurenblinde gebruikers. En als we na een tijdje honderden websites hebben laten testen (of eigenlijk exploreren) en beoordelen (leren), dan ontstaan er

algemene trends in verwachtingen waar de tool bij de volgende test rekening kan houden. Kennis en ervaring over honderden testobjecten, beschikbaar met één druk op de knop, is dit het einde van de professionele functioneel tester die we nodig hadden om zijn kunde en zijn jarenlange ervaring?!

Hoe haalbaar is dit concept?

Laten we weer even op adem te komen. Op dit moment zijn zelflerende en zelf-explorerende testtools nog fictief. Maar alle benodigde kennis en technieken bestaan op dit moment wel. Een belangrijke vraag is uiteraard of de investeringen in dit soort testtools opwegen tegen de resultaten. Met name het overnemen van het creatieve denkwerk van de menselijke tester klinkt nu nog heel futuristisch, maar de ontwikkelingen gaan erg snel.

De eerste stap zal zijn dat de testtools een deel van het verkennende werk doen en daarmee een basis leggen waarop de menselijke tester voort kan borduren. Maar voor de langere toekomst is het niet denkbeeldig dat dit soort zelflerende en zelf-explorerende tools op eigen initiatief de kwaliteit van een informatiesysteem vaststellen op een wijze die de betrokkenen voldoende zekerheid en vertrouwen geeft om het informatiesysteem in gebruik te nemen.

Wil jij samen met ons de toekomst verkennen?

We willen graag verder met het concept, samen met collega testers die hier graag over mee willen denken. De mogelijkheden lijken eindeloos. We denken al groot, en dan nog zullen we in de toekomst waarschijnlijk verbaasd worden. Maar er zijn ook genoeg obstakels om te overwinnen. Over dit kijkje in de toekomst is genoeg te bespreken! Heb je hier interesse in, laat het dan aan één van ons weten.