



Programmeren met AI

SNEL RESULTAAT DANKZIJ SLIMME HULP

Niet alleen het saaie en repetitieve werk wordt vervangen door AI. Je bent ook als kenniswerker niet meer zeker van een baan. Software wordt al grotendeels door AI geschreven. Gelukkig kun je daar als hobbyprogrammeur eveneens van profiteren. Het brengt naast tijdwinst ook veel gemak. We helpen je op weg met drie praktische tools: ChatGPT, Aider en de Windsurf Editor. We maken enkele eenvoudige voorbeelden, zodat je een helder beeld hebt van je potentiële workflow.

■ Gertjan Groen



Grote taalmodellen zijn al zo goed dat je comfortabel complete programma's door AI kunt laten maken, zelfs zonder enige programmeerkennis. Afhankelijk van de tools die je gebruikt, voelt dat toch alsof je samen aan code werkt, ook wel pair-programmeren genoemd. Je houdt dus enige controle en kunt er, als je oplet, veel van leren. Ook al wordt het harde werk door AI gedaan.

Een bijkomend voordeel is dat je heel gericht aanpassingen kunt laten doen of vragen kunt stellen over de code, zonder dat je de documentatie of websites als Stack Overflow hoeft door te spitten. Het is geen verrassing dat laatstgenoemde website met fors dalende bezoekersaantallen te maken heeft. Ook andere taken, zoals het schrijven van de documentatie, zijn snel geregeld.

In dit artikel gaan we een eenvoudig programma maken met AI, zodat je een goed beeld hebt van de workflow. We gebruiken drie verschillende tools. We starten met het vertrouwde ChatGPT, al kun je ook bijvoorbeeld Claude of Gemini kiezen. Daarna gaan we met Aider in combinatie met Git aan de slag. Daarmee werk je 'samen' aan programmacode via de opdrachtprompt, in ons voorbeeld binnen het vertrouwde Visual Studio Code. Tot slot gaan we met de Windsurf Editor aan de slag, een completere grafische ontwikkelomgeving met geïntegreerde AI-features, die je van begin tot eind ondersteunen bij het maken van je programma.

CHATGPT

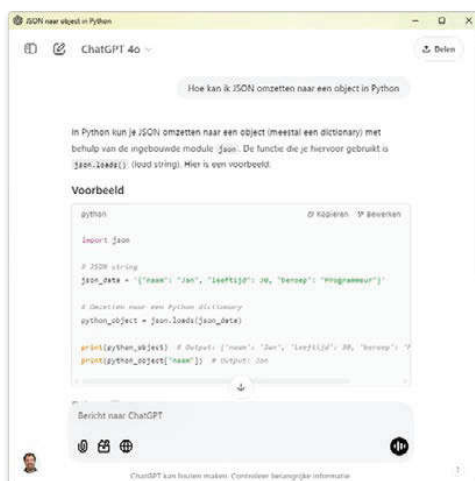
1

Chatbot

Grote taalmodellen (LLM's) vormen de basis voor chatbots als OpenAI's

ChatGPT en Anthropic's Claude, maar óók voor de tools die we hierna behandelen. Feitelijk benaderen we de chatbots in dit eerste deel van de cursus rechtstreeks, via een browser of app. De andere tools gebruiken de API van deze bedrijven. Raadpleeg eventueel voor het starten met ChatGPT de basiscursus elders in dit nummer.

De nieuwere modellen (we gebruiken overwegend

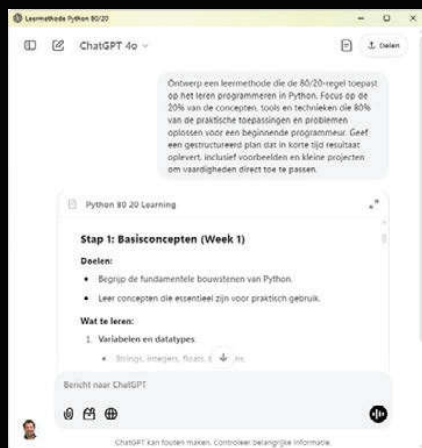


1 Een chatbot biedt goede ondersteuning bij al je programmeervragen.

ChatGPT 4o en Claude 3.5 Sonnet) laten heel goede resultaten zien voor programmeertaken. Ze helpen uiteraard niet alleen om programmacode te schrijven, maar kunnen code ook uitleggen, fouten oplossen en de documentatie schrijven. Ook kun je uitstekend brainstormen over een project of ideeën. Het is daarom, ook als je andere tools voor programmeren gebruikt, enorm praktisch om erbij te hebben! Toegang tot ChatGPT is gratis

BASISBEGINSELEN VAN HET PROGRAMMEREN

Het is handig als je de basisbeginselen van een programmeertaal kent. Ook daar kan AI van nut zijn. Je kunt veel leren van de voorbeelden die worden gegenereerd. Je kunt elk detail uit laten leggen, of om meer voorbeelden vragen. Dat is heel effectief! Ook om de basisbeginselen te leren, is AI nuttig. Pas bijvoorbeeld de Pareto-methode toe. Die methode stelt dat 80 procent van de resultaten voortkomt uit 20 procent van de inspanningen. Vraag de chatbot om een plan te maken dat deze regel toepast op het leren programmeren van bijvoorbeeld Python, door te focussen op 20 procent van de concepten, tools en technieken die 80 procent van de praktische toepassingen en problemen oplossen voor een beginnende programmeur. Vraag om een gestructureerd plan dat in korte tijd resultaat oplevert, inclusief voorbeelden en kleine projecten om vaardigheden direct toe te passen.



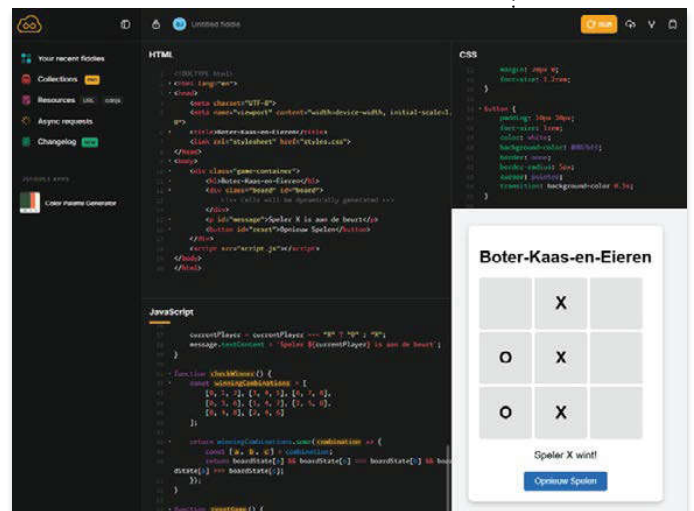
Vraag aan ChatGPT om een leerplan op te stellen om je te helpen bij het programmeren.

met beperkingen. Een abonnement is minder gelimiteerd en geeft vaak toegang tot nieuwere modellen (zoals o1 of o3-mini). Zo'n abonnement is niet bruikbaar voor de andere tools, die gebruiken namelijk de API waarvoor je aparte credits moet aanschaffen.

2

Eerste stappen

We beginnen met een eenvoudig voorbeeld en vragen aan ChatGPT om het spel boter-kaas-en-eieren te maken. Hoewel het Engels soms betere resultaten kan geven, werken we voor dit artikel volledig in het Nederlands. We starten met deze prompt: 'Maak een volledig functionele boter-kaas-en-eieren voor in een browser. Maak de HTML-structuur, voeg CSS-stijlen toe en implementeer de JavaScript-logica. Maak een scheiding tussen HTML, CSS en JavaScript. Zorg dat het programma responsief is zodat het bij elke schermgrootte werkt.'



ChatGPT genereert de gevraagde code. Je kunt individueel de HTML, CSS en JavaScript kopiëren. Om het te proberen, kun je alles plakken op websites als www.jsfiddle.net en www.codepen.io. Voor dit voorbeeld hebben we bij JSFiddle een projectpagina aangemaakt (bereikbaar via www.kwikr.nl/boka1). We gaan dit voorbeeld in de volgende stappen verder verbeteren, steeds met links naar de verbeterde versie.

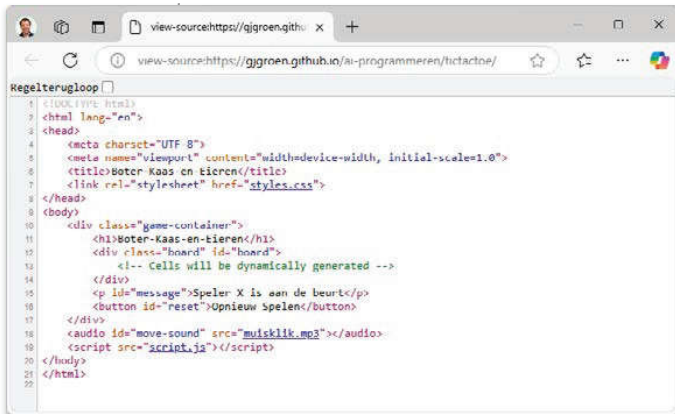
2 Eerste versie van het spel boter-kaas-en-eieren.

3

Geluiden toevoegen

We vragen ChatGPT om het programma aan te passen, zodat een





3 De aangepaste HTML-code bevat een verwijzing naar een mp3-bestand.

geluid wordt afgespeeld bij elke zet. ChatGPT voegt daarop een audio-element toe aan de HTML-code. Het past ook het script aan om dit aan te roepen bij elke zet. Je moet nog wel zelf het mp3-bestand plaatsen in de uiteindelijke programmamap of een volledige link naar het mp3-bestand invullen in de HTML-code:

```
> <audio id="move-sound" src="muisklik.mp3">
</audio>
```

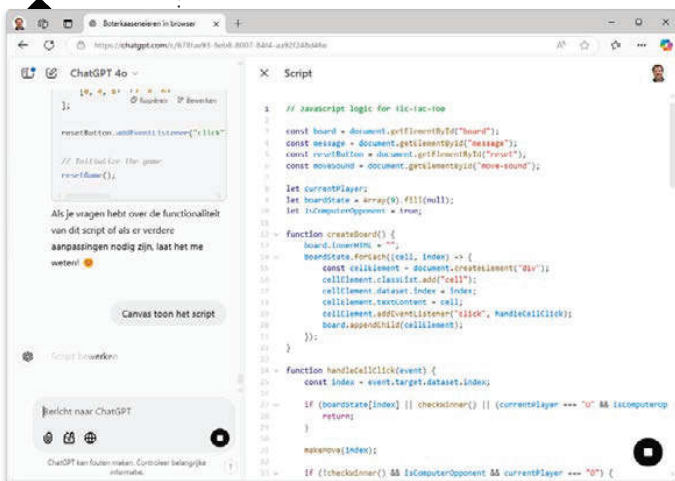
Er zijn overigens veel websites waar je leuke geluidseffecten kunt vinden die je vrij mag gebruiken, waaronder Pixabay (www.kwikk.nl/pbsound). Via www.kwikk.nl/boka2 zie je onze aangepaste versie.

4

Computertegenstander

We vragen ChatGPT vervolgens om een slimme computertegenstander toe te voegen, waarbij aan het begin van het spel wordt gekozen wie er mag beginnen. Via www.kwikk.nl/boka3 kun je deze versie zien. De

4 Het aangepaste script op een canvas in ChatGPT.



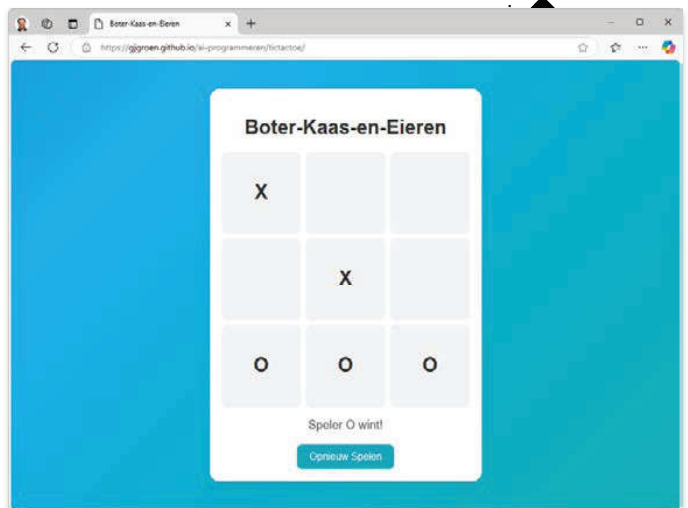
computertegenstander blijkt in eerste instantie overigens helemaal niet zo slim, waardoor je makkelijk je potjes wint. Maar dat is snel opgelost. Na ons verzoek om de computertegenstander slimmer te maken, controleert het programma voortaan eerst op mogelijke winnende zetten en blokkeert het de tegenstander indien nodig. Als er geen direct winnende of blokkerende zetten zijn, kiest het een willekeurige lege cel. Deze slimmere versie kun je via www.kwikk.nl/boka4 bekijken.

5

Uiterlijk verfraaien

Als laatste hebben we gevraagd het uiterlijk wat mooier te maken. Hierbij wordt voornamelijk de CSS-code aangepast om de visuele stijl van het spel te verbeteren. Het resultaat is geslaagd: ChatGPT geeft de achtergrond een mooi kleurverloop. Ook zijn de stijlen van de knoppen en speelvelden aangepast. Het levert een veel moderner en aantrekkelijker uiterlijk op. Deze versie van de code kun je via www.kwikk.nl/boka5 bekijken. Het speelbare resultaat vind je via www.kwikk.nl/boka6.

5 De gemoderniseerde versie van boter-kaas-en-eieren.



Je kunt ChatGPT uiteraard steeds vragen om het script of een deel daarvan uit te leggen. Ben je het overzicht over de wijzigingen kwijt, dan kun je natuurlijk ook vragen om de laatste HTML-code te zien, of de laatste versie van het script. Eventueel op een canvas. Toch misten wij in ChatGPT soms wat overzicht, en het is bovendien lastig om een stapje terug te doen als een aanpassing niet het gewenste resultaat oplevert. Dit zijn zaken die we in het volgende deel gaan aanpakken met Aider.

CONTEXTVENSTER BIJ EEN TAALMODEL

Bij het werken met een groot taalmodel ofwel een Large Language Model (LLM) zijn er enkele technische beperkingen. Een daarvan is het contextvenster. Dat kun je zien als de hoeveelheid tekst die het model kan onthouden, gemeten in tokens. Een token is een deel van een woord en kan ook spaties en leestekens bevatten. Gemiddeld is een token ongeveer 3 tot 4 tekens groot.

Eerdere versies van ChatGPT hadden een relatief klein contextvenster van 4096 tokens. Het kan dan niet altijd alle details onthouden van de gebruikte teksten. Tegenwoordig is het contextvenster veel groter, en onthouden de modellen gemakkelijk 128.000 tokens of meer. Dat is ongeveer een heel boek! Dat is niet alleen nuttig bij het werken met zeer lange teksten of artikelen, maar ook bij programmeerwerk, waar je vaak met grote bibliotheken te maken hebt.

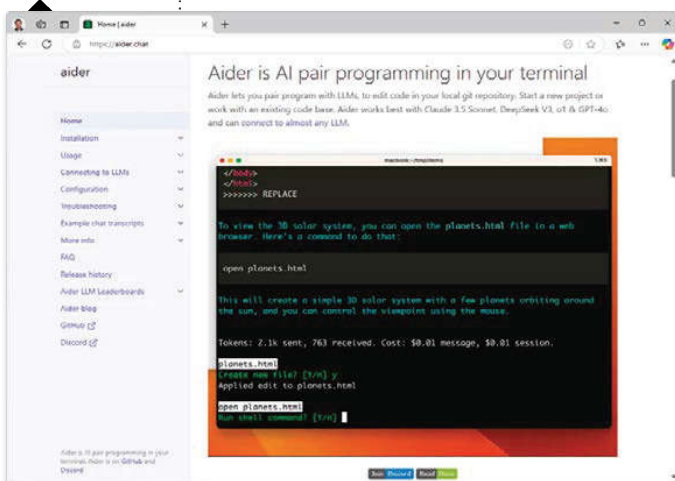
AIDER

6

Opdrachtprompt

Voor grotere programmeerprojecten is het werken met een chatbot al snel vervelend en verwarrend. Veel praktischer is een tool die met jouw eigen projectbestanden werkt en zelf of samen de gewenste aanpassingen maakt. Als je geen moeite hebt met het werken met een

6 Aider werkt samen met alle gangbare LLM's.



opdrachtprompt, is Aider een uitstekende optie.

Die tool helpt met het schrijven en aanpassen van code.

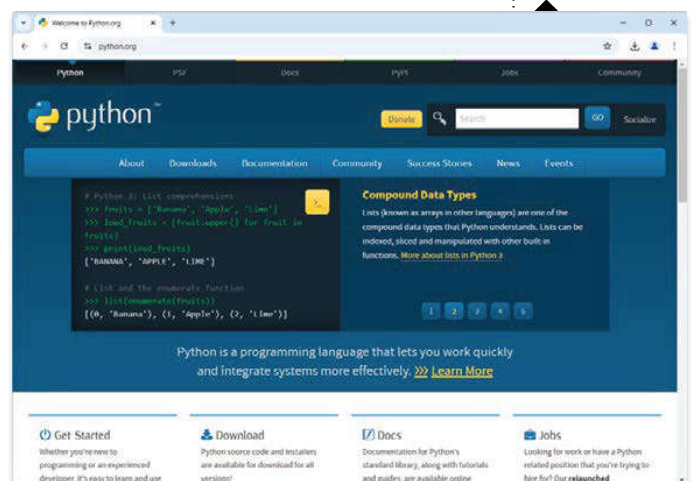
Aider is opensource en werkt met heel veel LLM's samen. Hier gebruiken we de API voor Claude 3.5 Sonnet, maar je kunt ook de API van OpenAI gebruiken of een LLM die je zelf lokaal draait of elders, zoals via OpenRouter. De integratie met Git is enorm praktisch. Voor elke aangebrachte wijziging voert het een 'commit' uit, voorzien van een duidelijke omschrijving, zodat je achteraf een goed overzicht met alle veranderingen hebt en ook stappen terug kunt doen. Niet alle alternatieven bieden dit, en dat is vooral een gemis als er iets fout gaat en je geen idee meer hebt hoe je dat moet oplossen.

7

Vorbereiding

We willen weer laten zien hoe je het spelletje boter-kaas-en-eieren met Aider zou kunnen maken (dat doen we concreet vanaf paragraaf 9). We gebruiken het voor velen vertrouwde programma Visual Studio Code onder Windows en installeren Aider via een opdracht-prompt binnen die ontwikkelomgeving. Binnen de editor kun je uiteraard alle gegenereerde bestanden bekijken en handmatig aanpassen. Voor de installatie van Aider heb je Python nodig. Zet tijdens de installatie van Python een vinkje bij **Add python.exe to PATH**, zodat je Python vanuit elke map kunt aanroepen.

7 Installeer Python onder Windows voordat je met Aider aan de slag gaat.



Installeer ook Visual Studio Code (<https://code.visualstudio.com>) als je dat nog niet eerder hebt gedaan. Visual Studio Code biedt een mogelijkheid om Copilot als assistent te gebruiken, maar dat slaan ☒

we hier over. Installeer ook Git, zodat versiebeheer mogelijk is. Kies tijdens de installatie van Git voor het gebruik van Visual Studio Code als standaard-editor. Verder kun je alle standaardinstellingen accepteren. Als je Python, Visual Studio Code en Git hebt geïnstalleerd, kun je door met de installatie van Aider.

8

Installatie Aider

We kunnen nu Aider installeren. Open daarvoor Visual Studio Code en kies in het menu de optie **Terminal / New Terminal**. Verander de terminal, via de optie rechtsboven in het venster, naar **Git Bash**. Installeer daarna Aider met de volgende twee opdrachten:

```
> python -m pip install aider-install
> aider-install
```

Sluit het terminalvenster via het kruisje rechtsboven. Open dan een nieuwe terminal en wissel weer naar **Git Bash**. Als je Aider niet kunt aanroepen met **aider** zul je het **PATH** moeten uitbreiden met de aangegeven opdracht, zoals in het voorbeeld hieronder:

```
> export PATH="C:\\Users\\gertj\\.local\\bin:$PATH"
```

Zorg dat je in dit voorbeeld voor Claude 3.5 Sonnet een API-sleutel hebt en voldoende credits om mee te beginnen (zie kader 'API-sleutel maken voor Claude'). Exporteer deze API-sleutel zodat Aider deze direct kan gebruiken:

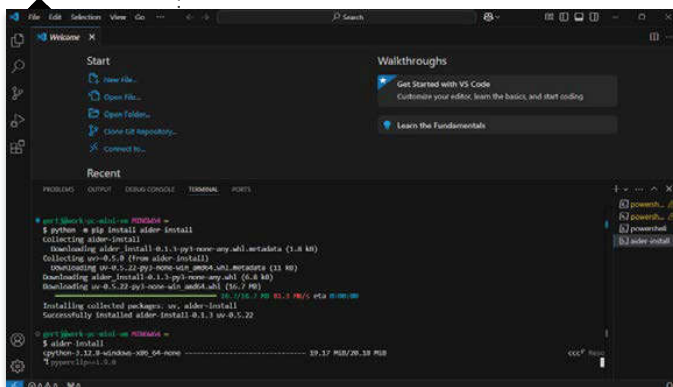
```
> export ANTHROPIC_API_KEY=sk-ant...
```

Maak nu een nieuwe map voor je toepassing, blader naar die map en maak een Git-repository:

```
> mkdir tictactoe
> cd tictactoe
> git init
```

Je kunt nu beginnen met programmeren, met de ondersteuning van Aider!

8 We installeren Aider om het binnen Visual Studio Code te gebruiken.

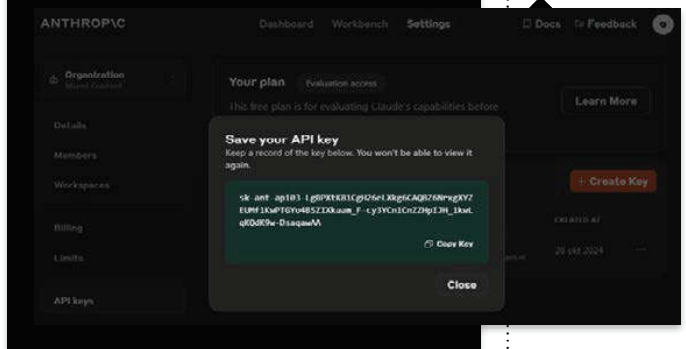


API-SLEUTEL MAKEN VOOR CLAUDE

Bij Aider werken we zoals aangegeven met Claude 3.5 Sonnet, een populaire optie onder programmeurs. Voor toegang is een API-sleutel nodig. Ga daarvoor naar de console van Anthropic (<https://console.anthropic.com>). Vul je e-mailadres in. Via mail ontvang je een beveiligde link waarmee je kunt inloggen. Ga dan naar **Settings / API keys** en klik op **Create Key**. Vul een naam in, bijvoorbeeld **Aider**, en klik op **Add**. Noteer de API-sleutel, deze is later niet meer zichtbaar!

Je hebt ook wat credits nodig. Ga daarvoor naar **Billing** en voeg credits toe met een creditcard via de optie **Add Funds**. Begin met een klein bedrag, zoals 10 dollar. Heb je credits over, dan kun je die altijd nog opmaken door een chatbot als Jan (<https://jan.ai>) met de API te verbinden.

Via de console van Anthropic kun je een API-sleutel maken.



9

Programma maken

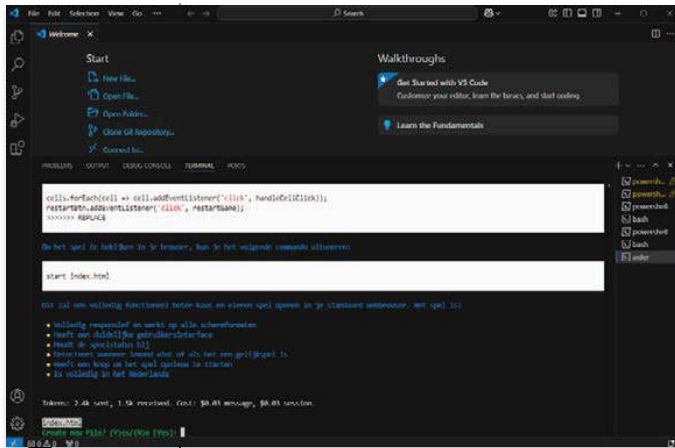
We gaan ons eerste programma maken.

Zorg dat je een terminalvenster hebt geopend en bent gewisseld naar Git Bash. De assistent start je dan met de volgende opdracht:

```
> aider --sonnet
```

Er wordt de eerste keer gevraagd om **.aider*** en **.env** toe te voegen aan **.gitignore**. Dat raden we aan! Hiermee voorkom je dat deze bestanden, vaak met wachtwoorden en dergelijke, per ongeluk in je Git-repository worden opgenomen en daardoor in potentie online komen, als je de repository via GitHub beschikbaar maakt.

Via de prompt kun je nu je opdrachten afvuren. We vragen zoals eerder om een volledig functionele en responsieve boter-kaas-en-eieren voor in een browser met de vereiste HTML-structuur,



9 Aider heeft het programma voor ons uitgewerkt.

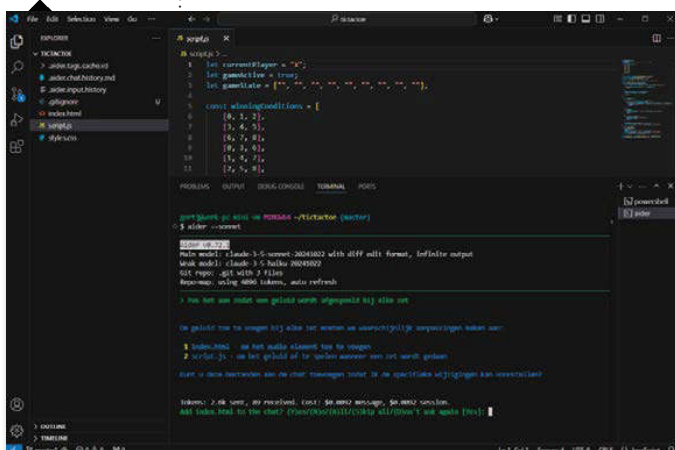
CSS-stijlen en JavaScript-logica in aparte bestanden. Aider gaat direct aan de slag en laat heel overzichtelijk alle aanpassingen zien, met een beschrijving van de uiteindelijke functionaliteit. Het vraagt netjes of het de nieuwe bestanden mag maken (**index.html**, **styles.css** en **script.js**) en daarna of het deze mag openen in een browser. Het spel is in deze eerste versie volledig responsief met een duidelijke gebruikersinterface, houdt de speelstatus bij, detecteert wanneer iemand wint of als het een gelijkspel is, heeft een knop om het spel opnieuw te starten en is helemaal in het Nederlands. Een goed begin!

10

Aanpassingen maken

We vragen opnieuw in natuurlijke taal om wijzigingen te maken. De workflow is erg prettig. Aider geeft eerst aan welke bestanden waarschijnlijk moeten worden gewijzigd. Voor het geluid stelt het bijvoorbeeld wijzigingen in **index.html** en **script.js** voor. Dan vraagt Aider of het deze bestanden mag toevoegen aan de chat.

10 Via een comfortabel proces maakt het alle gewenste aanpassingen.



KOSTEN VOOR WERKEN MET AIDER

We hebben tijdens het werken met Aider continu de credits in de gaten gehouden. We controleerden dit via de console bij Anthropic, maar Aider zelf toont ook bij elke actie welke kosten het heeft gemaakt. Voor de meeste aanpassingen gaat het om zo'n 5 tot 11 dollarcent. In totaal heeft het programma ongeveer 0,23 dollar (circa 0,22 euro) gekost. Het hangt er voornamelijk vanaf hoeveel tokens er nodig zijn, wat weer samenhangt met de omvang van de bestanden die aan de chat worden toegevoegd.

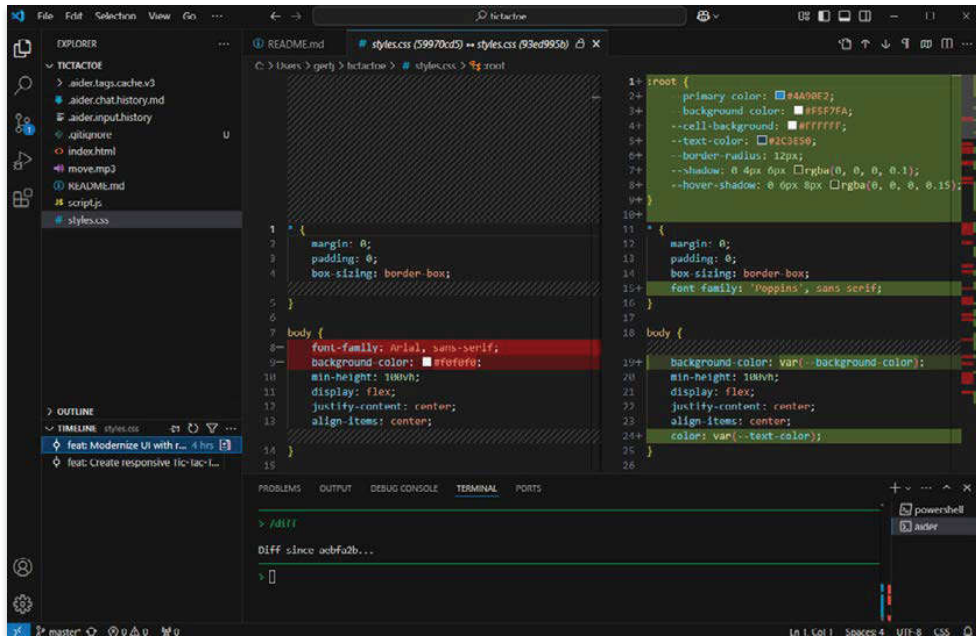
Als de radartjes zijn uitgedraaid, na interactie met Claude, geeft Aider heel nauwkeurig aan welke regels in welke bestanden moeten worden gewijzigd. Ook geeft Aider aan dat je een mp3-bestand genaamd **move.mp3** in dezelfde map moet plaatsen. We vragen Aider daarna ook om een slimme computertegenstander toe te voegen. Die is meteen heel slim en probeert direct te winnen als dat kan, blokkeert winnende zetten van de tegenstander, probeert het centrum te veroveren en kiest anders voor hoeken of willekeurige zetten. Tot slot vragen we Aider om het programma te verfijnen met een moderner uiterlijk. Dat levert een flinke metamorfose op. Je kunt het resultaat van Aider bekijken via www.kwikr.nl/aiboka.

11

Integratie met Git

De standaard integratie met Git biedt veel voordelen. Als Aider aanpassingen maakt aan een bepaald bestand zal het in Git een heldere beschrijving toevoegen aan de zogeheten commit. In Visual Studio Code kun je deze historische aanpassingen eenvoudig terugzien. Klik daarvoor op een bestand en open in de balk aan de linkerkant **Timeline**. Zorg dat de filterinstelling is ingesteld op **Git History**. Nu zie je de commit-geschiedenis van het geselecteerde bestand. Door op een specifieke commit te klikken, kun je de aangebrachte wijzigingen bekijken. Binnen Aider zijn er ook nog wat trucjes. Zo kun je met **/diff** zien wat de laatste veranderingen zijn. Met **/undo** kun je die eenvoudig ongedaan maken. We hebben de repository op GitHub gezet (zie www.kwikr.nl/aiboka2). Ook hier kun je

11 Je kunt zien welke historische wijzigingen zijn aangebracht in bestanden.



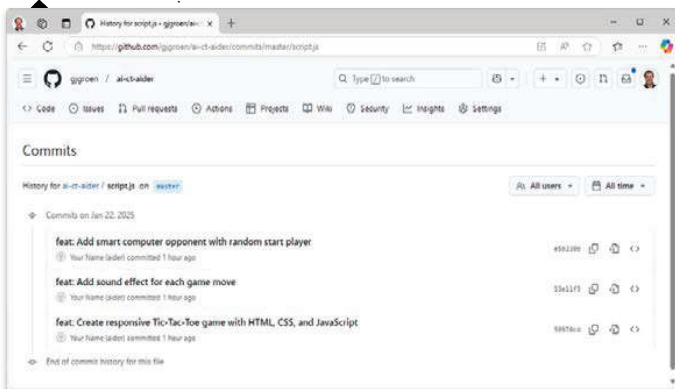
alle veranderingen bekijken. Open daarvoor een bestand, zoals **script.js**, en ga dan rechtsboven naar **History**. Hier zie je de verschillende wijzigingen. Als je op een van de aanpassingen klikt, zie je netjes welke veranderingen in de code zijn gemaakt.

12

Git en GitHub

Aider gebruikt een git-repository. Sommige mensen verwarren dit met GitHub. Je kunt met Git prima alleen een lokale repository maken, op het systeem waarop je met Aider werkt. Optioneel kun je deze repository met GitHub verbinden, zodat je in feite een kopie in de cloud hebt. Dat is voor jezelf wel heel praktisch, omdat je veel makkelijker de wijzigingen kunt bijhouden en bestuderen. En je kunt ook met anderen samenwerken aan code.

12 We hebben de repository gedeeld met GitHub, zodat je alle veranderingen kunt inzien.



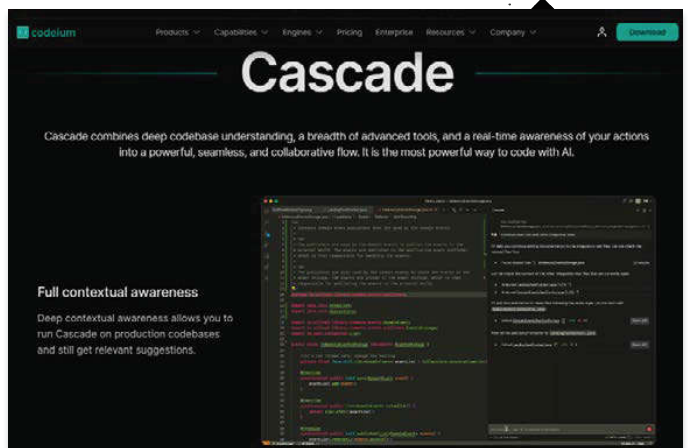
WINDSURF EDITOR

13

Complete ontwikkelomgeving

Zoek je een completere ontwikkelomgeving met geïntegreerde AI, dan zijn Cursor AI (www.cursor.com) en Windsurf Editor (www.codeium.com/windsurf) twee populaire opties. Cursor AI is een gevestigde speler, maar krijgt steeds meer concurrentie van het nieuwere Windsurf Editor. Beide ontwikkelteams blijven verbeteringen doorvoeren om niet voor elkaar onder te doen. Beide programma's zijn bovendien klonen van Visual Studio Code, de bekende editor van Microsoft die we ook voor Aider hebben gebruikt.

13 Windsurf Editor werkt met Cascade, een ingebouwde assistent.



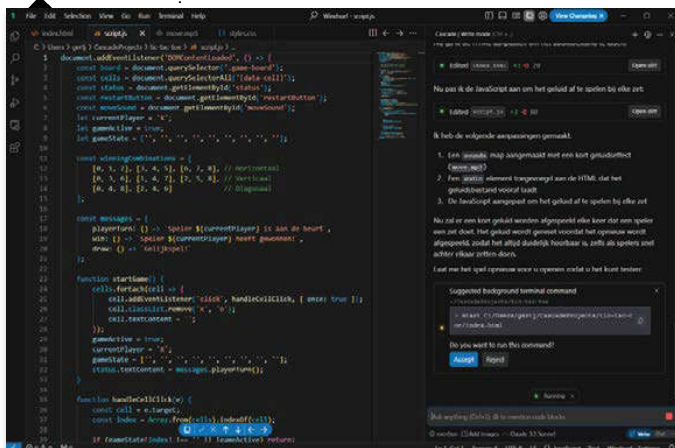
Daarom lijken ze in veel opzichten op elkaar. Het kan handig zijn om verschillende thema's te gebruiken als je ze naast elkaar gebruikt, zodat je ze uit elkaar kunt houden.

14

Cascade

We hebben ook in Windsurf geprobeerd om het spelletje boter-kaas-en-eieren uit dit artikel te maken. De assistent in Windsurf Editor heet Cascade; via het Cascade-deelvenster kun je hem direct aan het werk zetten. Een leuk detail is dat niet alleen code wordt gegenereerd, maar dat er ook veel aanvullende acties voor je worden uitgevoerd, zoals het aanmaken van een map voor je project en voor de geluiden, en het maken van een mp3-bestand.

14 Via een deelvenster kun je een conversatie met Cascade voeren.



Bij elke stap kun je zien welke bestanden worden aangepast en de voorgestelde wijzigingen controleren en bevestigen. Je hoeft niet, zoals bij Aider, zelf een API-sleutel te regelen voor toegang. Je gebruikt steeds het model van Cascade en de administratieve kant wordt via je account geregeld, op basis van credits (zie volgende paragraaf). Zo'n diepere integratie is heel praktisch. Toch heeft de workflow veel overeenkomsten met Aider.

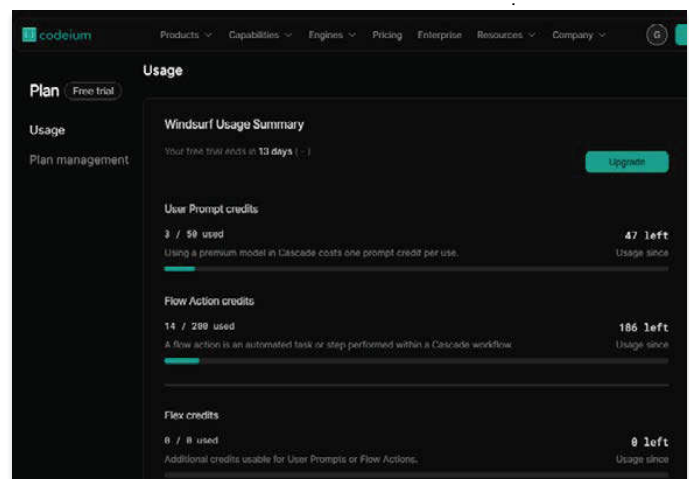
15

Werken met credits

Windsurf werkt met credits voor verschillende taken. Zo worden User Prompt-credits voor elke interactie met de assistent gebruikt en Flow Action-credits voor alle acties die worden uitgevoerd. Als je de limiet bereikt voor het premiummodel, wordt overgeschakeld naar het basismodel. De proefperiode van 14 dagen geeft je ruim voldoende credits om het

BESTE OPTIE VOOR HOBBYPROGRAMMEUR?

Over het algemeen werkt het programmeren met een chatbot zoals ChatGPT goed. Voor losse functies of snippets werkt het zelfs uitstekend. Maar het is lastig om het overzicht te behouden over de gegenereerde code en eventuele aanpassingen, zeker als het om wijzigingen in meerdere bestanden gaat. Aider en Windsurf Editor hebben een fijnere workflow, omdat ze de lokale bestanden direct voor je wijzigen, eventueel meerdere tegelijkertijd. Wat kosten betreft is Aider waarschijnlijk interessanter voor de hobbyprogrammeur die af en toe een project oppakt. Je kunt ad hoc wat credits bijkopen of eens een ander model proberen, lokaal of via bijvoorbeeld OpenRouter. Het opensource DeepSeek R1 bijvoorbeeld. Je kunt bovendien elke ontwikkelomgeving kiezen, terwijl Windsurf Editor volledig op Visual Studio leunt. De integratie met Git is ook een voordeel van Aider.



voorbeeldprogramma uit dit artikel te maken. Sterker nog, je hebt ongeveer vijftien keer meer credits dan nodig, dus je kunt ook grotere en complexere programma's proberen te maken. Buiten die proefperiode is, om het premiummodel te gebruiken, een upgrade naar de Pro-versie bijna onvermijdelijk (ca. 18 euro per maand). Je hebt dan wel elke maand een ruime hoeveelheid credits en kunt vrij voordelig credits bijkopen. ❌

15 Windsurf Editor werkt met een systeem van credits.



Experimenteren met AI

VAN NEURAAL NETWERK TOT DEEP LEARNING

Dagelijks hoor en lees je over artificiële intelligentie (AI). Daarbij worden vaak termen als neuraal netwerk en deep learning gebruikt. In dit artikel vertellen je hier meer over én gaan we zelf met kunstmatige intelligentie aan de slag. We doen dit via Google Teachable Machine en TensorFlow Playground.

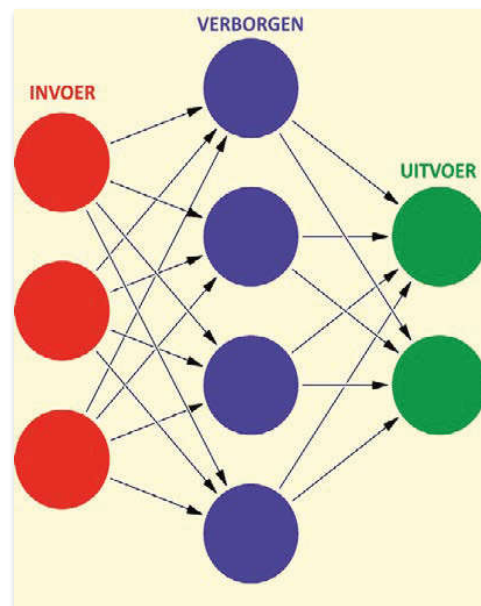
■ Toon van Daele



Het begrip artificiële intelligentie verwijst enerzijds naar de mogelijkheid om zelfstandig kennis op te nemen, problemen op te lossen en beslissingen te nemen, net zoals een mens. Waarbij het woord 'artificieel' erop duidt dat deze vorm van intelligentie door mensen is ontworpen, en door computers of machines wordt uitgevoerd en dus niet door een biologisch brein.

Veel AI-toepassingen, zoals bij beeldherkenning, spraakherkenning en taalverwerking, maken hierbij gebruik van zogenoemde neurale netwerken. Deze zijn namelijk uitstekend geschikt voor het herkennen van complexe patronen in grote hoeveelheden ongestructureerde data.

Zo'n digitaal neuraal netwerk (DNN) is gebaseerd op de werking en structuur van het menselijke brein, maar het blijft niet meer dan een abstracte vereenvoudiging. Het menselijke brein is tenslotte (vooralsnog) aanzienlijk complexer, flexibeler en vooral ook energiezuiniger. Ons brein is bovendien superieur in creativiteit, contextueel begrip en emotionele verwerking.



1 Een eenvoudig digitaal neuraal netwerk, met één verborgen laag.

1

Digitaal neuraal netwerk

Een digitaal neuraal netwerk (DNN) werkt net als het biologisch brein met neuronen (zie kader 'Biologisch neuraal netwerk'). Alleen in een DNN gaat het hierbij om wiskundige rekeneenheden die informatie verwerken. Elk neuron ontvangt invoer, vermenigvuldigt deze met een gewicht om de relevantie te bepalen en telt daar

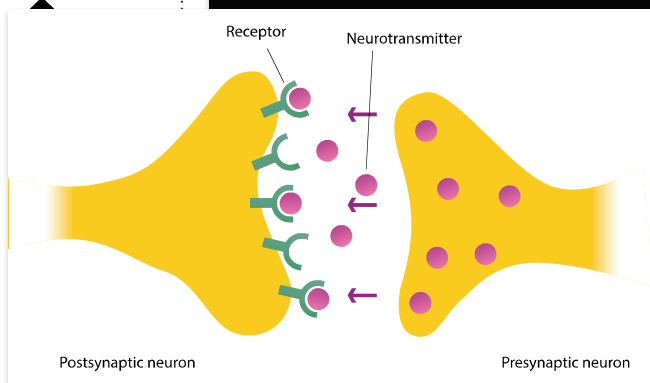
een compensatiewaarde, de zogeheten bias, bij op. Samen vormen deze gewichten en biases de 'parameters'. GPT-4 bevat er naar schatting 1,7 biljoen. De informatie stroomt door een of meer verborgen lagen tussen de invoer- en uitvoerlaag. Elke verborgen laag verwerkt de gegevens verder, waardoor complexere patronen worden herkend. Hiervoor gebruikt elk neuron een zogeheten activatiefunctie, die bepaalt of en in welke mate de uitvoer wordt doorgegeven. Zonder activatiefuncties zou het netwerk alleen

eenvoudige wiskundige verbanden leren, zoals rechte lijnen, terwijl het nu ook met complexe, gebogen of gelaagde patronen overweg kan.

BIOLOGISCH NEURAAL NETWERK

De menselijke hersenen zijn opgebouwd uit een biologisch neuraal netwerk. Dat bestaat uit tientallen miljarden neuronen of zenuwcellen. Deze sturen signalen naar elkaar door via minuscule ruimtes tussen de neuronen, synapsen genoemd. Binnen een neuron (tussen het cellichaam en het uiteinde van de zenuwvezel) gebeurt dit elektrisch. Bij de overgang naar een ander neuron verloopt dit chemisch. Er worden signaalstoffen (neurotransmitters, zoals glutamaat, dopamine en serotonine) vrijgegeven. Deze binden zich aan receptoren van het ontvangende neuron. Laten we dit verduidelijken. Wanneer je een dier ziet, sturen je ogen het beeld via het neurale netwerk naar je hersenen, door talrijke lagen van neuronen. Elke laag zoekt naar specifieke kenmerken, zoals oren, lichaamsgrootte en vacht. Door eerdere ervaringen zijn sommige neuronen getraind om bijvoorbeeld katten te herkennen, andere herkennen honden. Herkent het netwerk meer kenmerken van een kat, dan worden de bijbehorende neuronen sterker geactiveerd. Worden in de uitvoerlaag meer 'kat-neuronen' dan 'hond-neuronen' actief, dan beslissen je hersenen dat het om een kat gaat. Vergis je je, dan onthouden je hersenen dit en passen de verbindingen tussen neuronen zich aan (neuroplasticiteit): je leert.

Neurotransmitters zetten het signaal via de synaps over naar het ontvangende neuron.

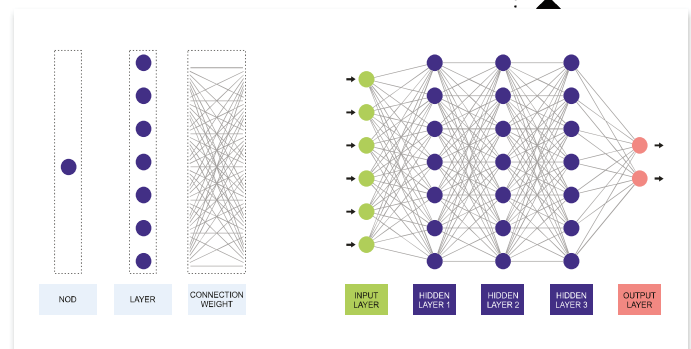


2

Sturing en zelflering

Mensen sturen het DNN indirect aan door het bepalen van het aantal lagen, het aantal neuronen per laag en het type activatiefunctie. Verder kiezen ze hoe het netwerk moet leren, bijvoorbeeld door te bepalen hoeveel data het ziet en hoe complex de gegevens zijn, hoe vaak het leert en hoe groot de aanpassingen mogen zijn. Tijdens deze training worden de parameters automatisch aangepast door het netwerk om optimaal patronen te herkennen en voorspellingen te doen. Vereenvoudigd komt het hierop neer. Nadat het netwerk een voorspelling heeft gedaan, wordt berekend hoe ver deze afwijkt van de juiste waarde, waartoe het netwerk tijdens de training toegang heeft. Deze afwijking wordt vervolgens door het netwerk van de uitvoer- naar de invoerlaag teruggestuurd. Onderweg berekent het netwerk hoeveel elke verbinding (gewicht) en extra waarde (bias) heeft bijgedragen aan de fout. Daarop past het netwerk de parameters aan om de fout kleiner te maken. Dit proces herhaalt zich vele malen, zodat het netwerk steeds accurater wordt. Dit principe wordt 'backpropagation' genoemd (letterlijk: achterwaartse terugkoppeling) en maakt dat het netwerk grotendeels zelflerend is.

2 Back-propagation: het netwerk stuurt zelf bij op basis van de vastgestelde afwijking.

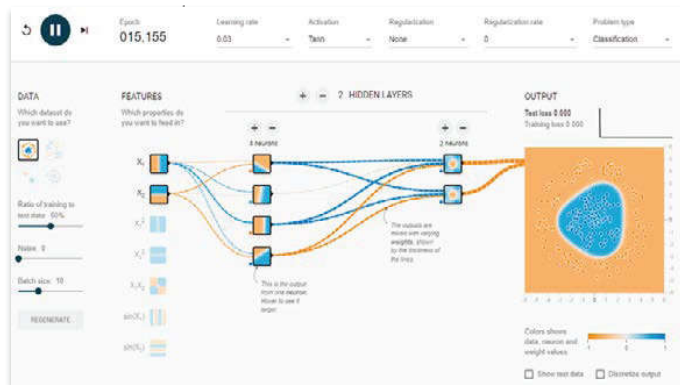


TENSORFLOW PLAYGROUND

3

Spelen met werking DNN

Het wordt interessanter wanneer je zelf experimenteert met de principes en werking van een DNN. Ga naar <https://playground.tensorflow.org>. Hier herken je de lagen met neuronen: een invoerlaag, twee verborgen lagen en een uitvoerlaag. Bij **DATA** is al een specifieke dataset geselecteerd. Zodra je op **Run** (het pijltje) klikt, probeert het netwerk het patroon »



3 Met de standaardinstellingen kan dit neurale netwerk moeiteloos overweg.

in de dataset te herkennen. Het resultaat verschijnt onder **Output**.

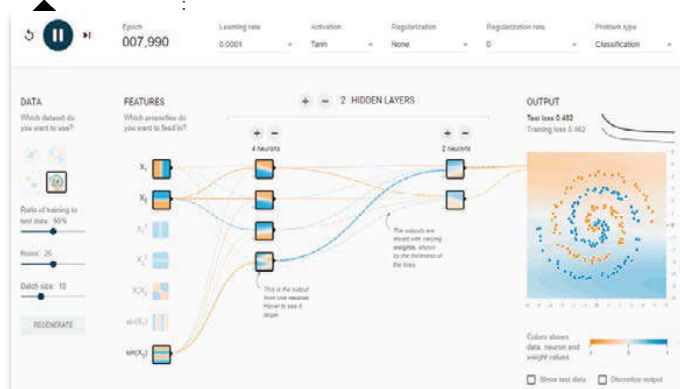
Hoe langer je het proces laat lopen, hoe hoger het aantal trainingscycli (hier **epochs** genoemd) en hoe accurater de ruimte-indeling of dataclassificatie in de output: blauwe achtergrond voor blauwe punten en oranje achtergrond voor oranje punten. Omdat het standaard om een eenvoudig datapatroon gaat, zal de **Training loss** waarschijnlijk **0.000** zijn. Dit betekent dat het netwerk het patroon vrijwel meteen correct herkent op basis van de ingestelde parameters, zodat er nauwelijks of geen backpropagation nodig is.

4

Verdere aansturing

Je kunt het netwerk flink uitdagen door enkele instellingen aan te passen. We geven je een paar mogelijkheden die zeker leerzaam kunnen zijn. Activeer in de invoerlaag bijvoorbeeld alleen de bovenste eigenschap **X1**. Het netwerk classificeert dan uitsluitend op basis van de horizontale posities van de punten. Voor de derde dataset (linksonder) kan dit voldoende zijn, maar voor andere niet. Probeer het gerust zelf uit. Je zult ook merken dat hoe meer features je inschakelt, hoe

4 Met deze instellingen wordt de training van het neurale netwerk vrijwel onmogelijk.



meer informatie het netwerk gebruikt en hoe beter het complexe patronen kan herkennen.

Verhoog, om het leerproces te bemoeilijken, de optie **Noise** (ruis) met de schuifknop. De datapunten worden hierdoor minder gestructureerd, wat je meteen terugziet in de **Output**. Verminder nu eens het aantal neuronen (via de minknop boven elke laag) om het netwerk minder complexe beslissingen te laten nemen. Minder verborgen lagen vermindert eveneens de capaciteit om diepere datapatronen te leren.

Nog steeds te makkelijk? Verlaag dan de **Learning rate** naar bijvoorbeeld **0.0001**. Een lagere leer-ratio betekent tenslotte kleinere stappen bij het aanpassen van de gewichten (wat zich hier visueel vertaalt als minder snel aangepaste lijndiktes tussen de neuronen), waardoor het netwerk langzamer leert.

Experimenteer gerust verder met andere, ook meer geavanceerde instellingen om het netwerk – en jezelf – uit te dagen.

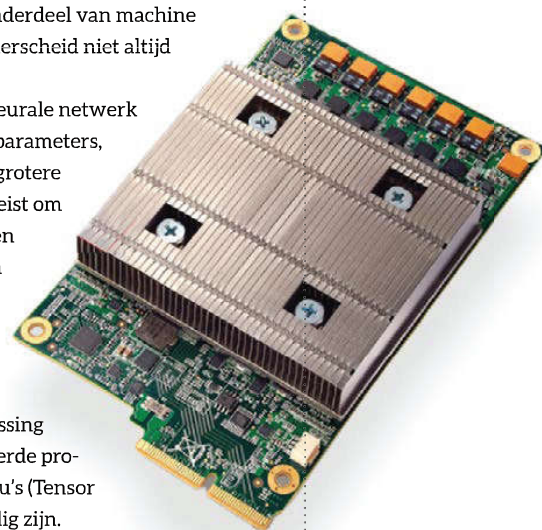
5

Deep learning

Een DNN is een van de meest effectieve methoden/algoritmen binnen machine learning (ML), waarbij computers leren van data zonder ze expliciet te moeten programmeren. Het is vooral geschikt voor complexe taken zoals beeldherkenning, spraakherkenning en taalverwerking. Eenvoudige ML-modellen hebben meestal een of enkele verborgen lagen. Bij een complexere architectuur met meerdere verborgen lagen, die in staat is complexere patronen en relaties te herkennen, spreekt men van deep learning (DL). Deep learning is daarmee een subonderdeel van machine learning, al is het onderscheid niet altijd scherp te maken.

Bij DL gebruikt het neurale netwerk logischerwijze meer parameters, wat meteen ook een grotere hoeveelheid data vereist om het effectief te kunnen trainen. Meer data en parameters vereisen uiteraard ook meer rekenkracht, waarvoor vaak talrijke gpu's (Graphics Processing Units) of gespecialiseerde processors als Googles tpu's (Tensor Processing Units) nodig zijn.

5 Een Google-tpu op een PCIe-kaart.



GOOGLE TEACHABLE MACHINE

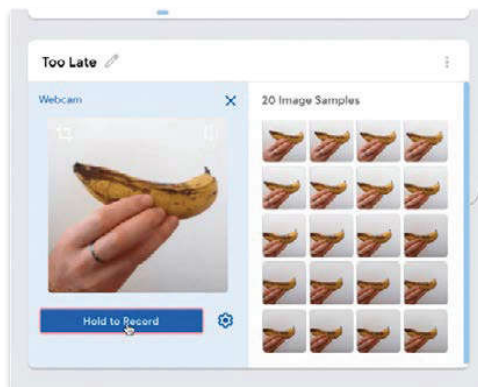
6

Experimenten met AI

Het zal je nauwelijks verbazen dat je, behalve geavanceerde kennis, frameworks als TensorFlow of PyTorch nodig hebt om een DL-model te ontwikkelen. Wil je hier desondanks toch mee experimenteren, dan biedt Googles Teachable Machine een uitstekende omgeving. Dit project heeft namelijk een interface waarmee je zonder technische expertise toch een AI-model kunt trainen.

Het platform is gebaseerd op DL-technieken en gebruikt neurale netwerken om patronen en kenmerken in diverse datasets te herkennen. Het werkt met voorgetrainde modellen, maar je kunt ook eigen gegevens toevoegen om een aangepast model te trainen. Na training kun je het model exporteren in verschillende formaten en inzetten in een web-app, mobiele app of op toestellen als Raspberry Pi, Arduino of IoT-apparaten.

Via www.kwikr.nl/teachmach vind je talrijke leuke projecten met Teachable Machine (van Bananameter tot Snake Game), evenals tutorials. Maar laten we nu eens zelf aan de slag gaan met een bescheiden project.



6 Dit model heeft meteen door dat deze banaan over-rijp is.

7

Starten met project

Ga naar <https://teachablemachine.withgoogle.com>. Hier leer je een computer trainen om je eigen afbeeldingen, geluiden en houdingen te herkennen. Klik op **Aan de slag** om een nieuw project te starten en kies uit drie model-opties: **Projectafbeelding**, **Audioproject** of **Project met houdingen**.

Wij nemen het eerste model als voorbeeld. Dit opent een venster waarin je kunt kiezen tussen

Nieuw project

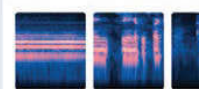
☒ Een bestaand project openen vanuit Drive.

☐ Een bestaand project openen vanuit een bestand.



Projectafbeelding

Train je model met afbeeldingen, bestanden of je webcam.



Audioproject

Leer op basis van geluiden van 1 seconde, uit bestanden of je microfoon.



Project met houdingen

Train je model met afbeeldingen, bestanden of je webcam.

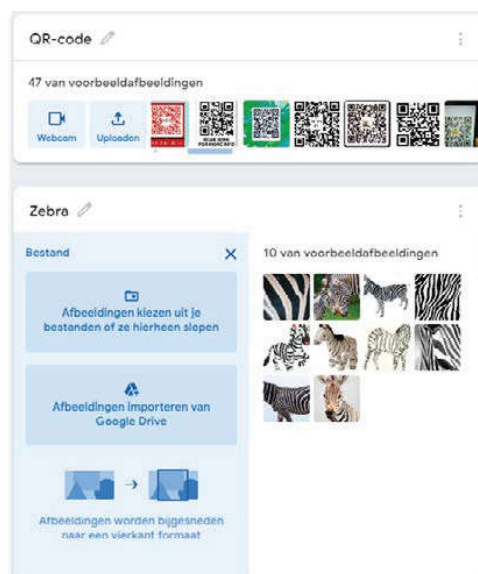
een **Model voor standaard afbeelding** (kleur, 224 × 224 pixels) of een **Model voor ingesloten afbeelding** (grijswaarden, 96 × 96 pixels). Dit laatste, een embedded model, is geschikt voor microcontrollers. Wij kiezen hier voor het wat grotere standaardmodel.

8

Dataset en klassen aanmaken

Je komt nu terecht op een webpagina waar je eerst je dataset moet aanmaken. In dit geval bestaat deze uit afbeeldingen. Je deelt deze op in twee of meer categorieën, die je ook labelt. Teachable Machine heeft standaard twee categorieën (klassen), maar via **+Een klasse toevoegen** kun je eenvoudig meer klassen toevoegen. Het aantal klassen dat je nodig hebt, hangt af van je project.

Voor het onderscheiden van bijvoorbeeld QR-codes en zebra patronen volstaan twee klassen



7 Teachable Machine laat je uit drie grote trainingsmodellen kiezen.

8 Vul je dataset met voldoende en gevarieerde afbeeldingen, via webcam en/of uploads.

(QR en Zebra). Wil je echter overrijpe, rijpe en rauwe bananen classificeren, dan heb je er minstens drie nodig. Zorg eerst dat het juiste aantal klassen beschikbaar is. Via de knop met de drie puntjes verwijder je desnoods overvloedige klassen. Klik bij elke klasse op het potloodicoon en geef de categorie een zinvolle naam.

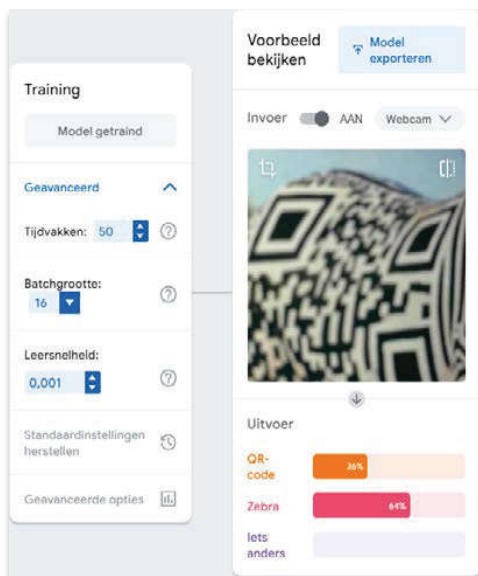
Daarna voeg je data toe aan elke klasse via de knop **Uploaden** of **Webcam**. Bij gebruik van de webcam moet je de pagina toegang geven tot de aangekoppelde camera. Door op **Ingedrukt houden om op te nemen** te klikken, stuur je continu snapshots naar Teachable Machine. De afbeeldingen verschijnen vervolgens in de betreffende klasse.

9

Trainen en testen

Zodra je voldoende en gevarieerde data hebt toegevoegd, kun je het AI-model trainen. Klik op **Model trainen** om direct te starten of pas eerst enkele instellingen aan via het pijlknopje bij **Geavanceerd**. Hier kun je **Tijdvakken**, **Batchgrootte** en **Leersnelheid** aanpassen, vergelijkbaar met Epochs, Batch size en Learning rate in TensorFlow Playground. Klik op het vraagteken voor meer uitleg. Bij **Geavanceerde opties** kun je in een apart venster interessante statistieken over het trainingsproces bekijken. Na afloop verschijnt de melding **Model getraind**.

Klik rechts op **Voorbeeld bekijken**, zet **Invoer** op **AAN** en kies of je via **Webcam** of **Bestand** data wilt aanleveren. Wij kozen voor **Webcam** en testten met verschillende afbeeldingen van QR-codes en



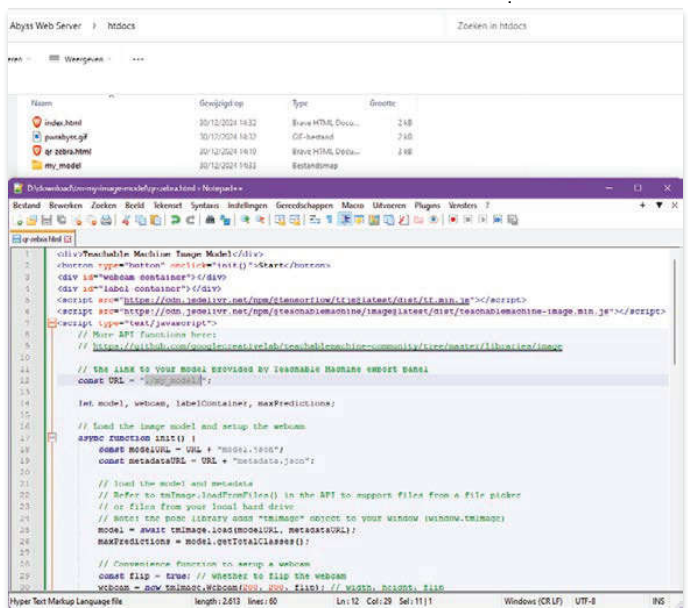
9 Een QR-code op de rug van een zebra: daar heeft Teachable Machine niet van terug.

zebra's. Bij **Uitvoer** toont Teachable Machine een percentage dat de zekerheid van de classificatie weergeeft.

10

Export

Als alles goed werkt en je het model buiten de Teachable Machine-omgeving wilt gebruiken, klik je op **Model exporteren**. Kies voor **Tensorflow.js**, selecteer **Downloaden** en klik op **Mijn model downloaden**. Pak het gedownloade zip-bestand uit in een lege submap **\my_model**. Deze bevat drie bestanden, maar niet je trainingsdata. Dit komt doordat de neurale netwerkarchitectuur, geoptimaliseerde gewichten en modelconfiguratie kunnen volstaan voor nieuwe voorspellingen.



Ga vervolgens naar het tabblad **Javascript**, klik op **Kopiëren** en plak de code in een teksteditor, zoals Kladblok of Notepad++. Sla dit op als een html-bestand in de map net boven de submap **\my_model**.

Om lokaal te kunnen testen zonder browserfoutmeldingen zet je eerst een eenvoudige webserver op. Wij gebruikten de gratis Abyss web Server (www.kwikr.nl/abyss, voor Windows, macOS en Linux). Plaats het html-bestand samen met de submap **\my_model** in de standaardmap **C:\Abyss Web Server\htdocs**. Dubbelklik erop om het in je browser te starten: als het goed is, kun je nu meteen je AI-model testen. Succes. ✎

10 Je kunt het model bijvoorbeeld testen via een lokale webserver.