

Po co komu dowód? (albo słów kilka o sztucznej inteligencji)

Drogi Czytelniku Majsterkowiczu! W tym odcinku Migawki wreszcie będzie ciekawie. Opiszemy, jak samodzielnie skonstruować prawdziwy zegarek mechaniczny!

Postawmy jednak sprawy bardzo uczciwie: nie mamy zielonego pojęcia, jak taki zegarek dokładnie działa. Nasza wiedza sprowadza się do bardzo prostych faktów: taki zegarek składa się z: kilku (a może kilkunastu?) zębatek o pewnej (zapewne nie za dużej) liczbie zębów; pewnej liczby metalowych rurek (o różnych średnicach i długościach); sprężyny zwiniętej spiralnie (także opisanej kilkoma parametrami); wskazówek, pokręteł, obudowy itp. itd.

Generalnie: cel w konfrontacji z naszą wiedzą wydaje się zupełnie nieosiągalny. Spróbujmy więc do sprawy podejść metodą tak zwaną *brutalnej siły*. To znaczy: spróbujmy (po kolei!) skonstruować wszystkie możliwe maszyny złożone z części opisanych wyżej. Już samych elementów (uwzględniając parametry) jest bardzo dużo, a do tego jeszcze możemy bardzo różnie ustawiać je wzajemnie w przestrzeni. Innymi słowy: ilość maszyn-kandydatek na działający zegarek jest astronomiczna. Większość z nich w ogóle nic nie robi, ALE przynajmniej jedna z nich będzie działała zgodnie z naszymi oczekiwaniami.

Nasza metoda jest więc następująca: buduj maszyny po kolei; każdą *testuj*, sprawdzając, czy aby nie *zbudował się* nam prawdziwy zegarek; jeśli test wypadnie pozytywnie – przerwij dalsze poszukiwania.

Nie da się zaprzeczyć, że powyższy algorytm opisuje poprawną metodę konstrukcji zegarka mechanicznego. Ma ona, oczywiście, ewidentne wady. Pierwsza jest taka, że metoda działa niesłychanie wręcz wolno. Druga, może nieco bardziej subtelna, bierze się z faktu, że maszynę uznajemy za poprawną, gdy ją (jak to określiliśmy) przetestowaliśmy. Można by długo dyskutować, co to właściwie znaczy i kiedy jakąś maszynę wolno nam zweryfikować jako Prawdziwy Zegarek Mechaniczny. Niemniej jednak – postawiony cel osiągnęliśmy!

Powyżej opisany sposób postępowania nie jest li tylko mało udanym żartem. Daje on pewne intuicje o tym, na jakiej zasadzie działa tak zwana sztuczna inteligencja, a bardziej konkretnie – uczenie maszynowe oparte o sieci neuronowe.

Opiszmy jakiś przykład. Proponuję rozważyć problem rozpoznawania cyfr napisanych odręcznie przez człowieka. Chcielibyśmy mieć algorytm, który potrafi to zadanie rozwiązywać automatycznie. Innymi słowy, chcemy stworzyć program komputerowy, który na wejściu dostanie zdjęcie kartki papieru, a na wyjściu określi, jaka cyfra została sfotografowana (upraszczamy problem do rozpoznawania jednego znaku).

Aby taki algorytm stworzyć, postąpimy analogicznie jak przy konstrukcji zegarka mechanicznego. To znaczy: skonstatujemy na początku, że nie mamy konkretnego planu, jak by się do tego zadania zabrać wprost. Jakaś tam wiedzę ogólną jednak mamy. Wiemy, że pewna bardzo skomplikowana struktura – mózg ludzki – radzi sobie świetnie z takim zadaniem. Niestety, nie umiemy zaimplementować w komputerze ludzkiego mózgu. Z drugiej strony, ludzki mózg wykonuje zadania znacznie bardziej skomplikowane niż rozpoznawanie cyferek, więc pewnie jakaś o wiele prostsza struktura, ale jednak do mózgu choć trochę podobna, ma szansę zadziałać.

Zakładamy więc, że nasz algorytm będzie działał na obraz i podobieństwo BARDZO uproszczonego mózgu.

Zacniemy od definicji klocków konstrukcyjnych. Będą nimi *sztuczne neurony*, które będziemy mogli dowolnie łączyć, tworząc sieć. Pojedynczy sztuczny neuron działa bardzo prosto. Na przykład w modelu McCullocha–Pittsa zakładamy, że może on mieć wiele wejść (ale jedno wyjście), którym przekazywane są jakieś liczby, a wyjście jest ich jakaś (bardzo prostą) funkcją. Nasz docelowy algorytm będzie zawierał opis sieci takich prostych neuronów wraz z opisem ich parametrów (np. opisem ich wewnętrznej funkcji). Samo obliczenie będzie polegać na tym, że wejściowe zdjęcie przerobimy na ciąg bitów, które będą z kolei wejściem do pewnego (ustalonego) zbioru neuronów. Następnie wszystkie neurony wewnętrzne dokonają obliczeń zgodnych ze swoim opisem i zastanym wejściem, aż neuron wyjściowy poda wynik. Oczywiście, cała sztuka polega na znalezieniu właściwej sieci i właściwego zbioru parametrów.

Sam algorytm poszukiwania właściwego algorytmu (sic!) mógłby działać następująco: buduj po kolei różne algorytmy (kandydatów); każdy kolejny testuj, aż uda się znaleźć jakiś działający poprawnie.

(W rzeczywistości działamy nieco sprytniej: wcale nie przeszukujemy kandydatów zupełnie losowo po kolei. Parametry kolejnych rozpatrywanych algorytmów są zależne od wcześniejszych testów. Np. przyjmujemy (heurystycznie), że gdy jakiś kandydat działa *już całkiem nieźle*, to staramy się dalej szukać w jego pobliżu, a więc nieznacznie modyfikując jego parametry.)

Dodatkowo musimy wyjaśnić, czym jest w tym przypadku testowanie. Otóż, aby to wszystko miało sens, musimy mieć (najlepiej bardzo dużą) bazę danych zdjęć cyfr napisanych odręcznie przez ludzi wraz z poprawną informacją, o jakie cyfry chodzi. Wówczas testowanie to po prostu weryfikacja, jak nasz algorytm poradziłby sobie z daną bazą danych.

To wszystko wyżej brzmi pewnie trochę surrealistycznie. A jednak – okazuje się, że gdy rozważane sieci są odpowiednio duże, nasze komputery odpowiednio szybkie, nasze heurystyki w wyborze kolejnych kandydatów trafione, a nasze bazy danych odpowiednio duże – to po prostu działa! W rozsądnym czasie potrafimy znaleźć algorytm, który radzi sobie nieźle z postawionym problemem. Więcej, opisana wyżej metoda działa zaskakująco dobrze z jeszcze trudniejszymi zadaniami, takimi jak rozpoznawanie przeszkód z obrazu kamerki samochodowej czy dobieranie właściwej reklamy do użytkownika serwisu internetowego.

To, co jednak powinno przerażać (szczególnie matematyka), to fakt, iż informatyk, który *znalazł* jakiś algorytm w stogu neuronowego siana, ani nie wie *tak naprawdę*, jak on działa, ani tym bardziej nie potrafi nic o nim udowodnić...

Tomasz KAZANA

PS. Drugi rozdział swojej książki „O Matematyku, rycerzu Gwiazdy Pitagorejskiej” nasz redaktor naczelny, Marek Kordos, kończy tak: *I z pewnym niepokojem należy uświadomić sobie, że dzisiejsza, prawie już dyrygująca światem informatyka też odstępuje od dowodzenia swoich algorytmów, zadowolając się przetestowaniem powstających programów.* Po przeczytaniu tych słów miałem, oczywiście, mieszanie uczucia. Że nie do końca, że nie zawsze. Ale w sztucznej inteligencji przynajmniej: do końca i zawsze.