

Archipelag sztucznej inteligencji. Część I

Ryszard Tadeusiewicz

Dlaczego „archipelag”?

Tytuł tego artykułu może budzić wątpliwości Czytelników. Sztuczna inteligencja? Wiadomo! Ale jakiś archipelag?

Już wyjaśniam.

Otóż sztuczna inteligencja tylko z nazwy jest dziedziną integralną, jak – nawiązując do tytułu miesięcznika – napędy albo sterowanie. W istocie sztuczna inteligencja to zbiór bardzo różnych metod, które ludzie wymyślili w tym celu, żeby maszyny lepiej zaspokajały ich potrzeby. Te metody w większości nie mają ze sobą nawzajem absolutnie nic wspólnego. Są od siebie odległe i nie ma łatwego sposobu przejścia od jednej z nich do innej. Pozwoliłem sobie porównać tę sytuację do archipelagu wysp (rys. 1).

W artykule omówimy niektóre z tych „wysp”, przy czym każdej z nich poświęcimy oddzielny rozdział. Fakt, że owe „wyspy” nie łączą się w sposób naturalny, nie wyklucza możliwości istnienia metod hybrydowych, opartych równocześnie na kilku z nich. Na przykład wzmiankowane dalej metody neuronowo-rozmyte opierają się na metodzie sieci neuronowych i na metodzie zbiorów rozmytych, są więc oparte na dwóch oddzielnych wyspach.

Tu drobna uwaga formalna: dla uproszczenia narracji nie będziemy dalej używali cudzysłowu przy słowie wyspa, ale zawsze będziemy mieli na myśli metaforyczne znaczenie tego słowa.

Wracając do wspomnianych metod hybrydowych, możemy je sobie wyobrażać na podobieństwo Kolosa Rodyjskiego, który w starożytności stał na dwóch wyspach u wejścia do portu na wyspie Rodos, z pochodnią, która pełniła rolę latarni morskiej (rys. 2).



Rys. 1. Przyjęta w tym artykule metafora zbioru metod sztucznej inteligencji jako izolowanych „wysp” tworzących archipelag



Rys. 2. Posąg Kolosa Rodyjskiego jako metafora hybrydowych metod sztucznej inteligencji

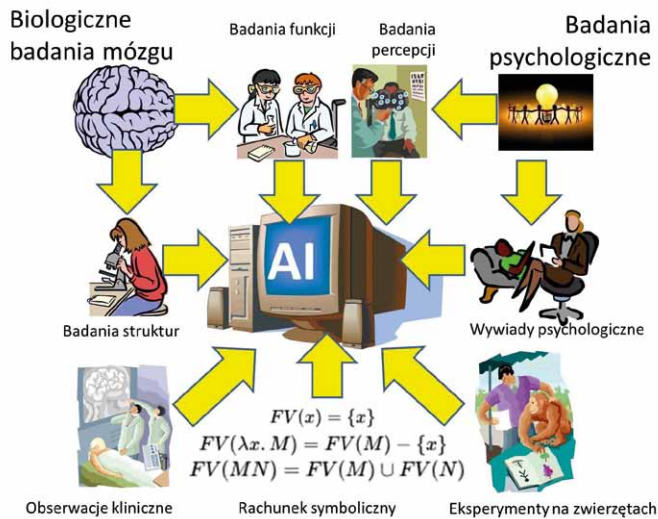
Zanim przejdziemy do meritum, potrzebna jest jeszcze jedna uwaga formalna. Otóż tekst tego artykułu został dość mocno oparty na rozdziale w książce, przywołanym w spisie literatury jako pozycja [1]. Oba teksty różnią się w wielu miejscach, ale jest też wiele świadomie użytych zapożyczeń.

Z czego się wywodzą metody sztucznej inteligencji?

Powodem tego, że sztuczna inteligencja jest tak wewnętrznie skomplikowana i niejednorodna, jest jej **geneza**. Ludzie, potrzebując narzędzi, które by im lepiej służyły, sięgali do różnych źródeł. Na rysunku 3 pokazano wybrane źródła wiedzy, z których czerpią wzory i inspiracje twórcy sztucznej inteligencji. W istocie jest tych źródeł znacznie więcej, ale pokazano te, które – jak się wydaje – wywarły największy wpływ na powstanie i rozwój omawianej tu dziedziny. W dodatkowym komentarzu do rysunku warto odnotować skrót **AI** (od *Artificial Intelligence*), używany dość powszechnie do oznaczania sztucznej inteligencji. Polski skrót **SI** (od Sztuczna Inteligencja) raczej się nie przyjął, zwłaszcza dlatego, że w technice i w naukach ścisłych skrót SI jest dość powszechnie kojarzony z jednolitym systemem znormalizowanych jednostek miar.

Pierwsze próby uzyskania inteligentnego zachowania maszyn opierały się na wykorzystaniu **rachunku symbolicznego** (rys. 3 – centralnie u dołu). O tym podejściu obszerniej będzie mowa w jednym z następných rozdziałów, więc tutaj nie wyjaśniamy szczegółów, tylko odnotowujemy hasło.

Drugie ważne podejście do tworzenia metod sztucznej inteligencji opierało się na wynikach badań mózgu (rys. 3 – lewy górny róg). Twórcy metod sztucznej inteligencji, wiedząc, że inteligencja człowieka rodzi się w mózgu – naśladują w swoich algorytmach struktury i funkcje jego elementów, tworząc i wykorzystując tak zwane **sieci neuronowe**. Także i ta grupa metod będzie dalej szerzej omawiana w jednym z dalszych rozdziałów.



Rys. 3. Źródła wiedzy wykorzystywane przy tworzeniu metod sztucznej inteligencji

Po prawej stronie u góry omawianego rysunku zasygnalizowane jest inne źródło wiedzy, na którym opiera się sztuczna inteligencja. Źródłem tym są **badania psychologiczne**. Nie ulega wątpliwości, że ludzkie procesy mentalne mogą być wzorem dla funkcjonowania narzędzi sztucznej inteligencji. Niektóre z tych procesów mogą być zbadane przy użyciu specjalistycznej aparatury (na przykład percepcja, zapamiętywanie i rozpoznawanie różnych wzorców), a inne – za pomocą wywiadów. Wprawdzie dalecy jesteśmy jeszcze od uzyskania pełnej wiedzy o naturze i przebiegu procesów kognitywnych w umyśle człowieka, ale nawet te fragmenty wiedzy, które udało się pozyskać, mogą być przydatne przy budowie systemów sztucznej inteligencji. Opis i dyskusja działania tych systemów będą przedmiotem dalszych rozdziałów.

Na omawianym rysunku są też jeszcze dwa źródła, z których sztuczna inteligencja czerpie wzory i inspiracje: są to doświadczenia na zwierzętach oraz obserwacje kliniczne pacjentów z różnymi dysfunkcjami mózgu. Tych źródeł nie będziemy jednak tutaj szczegółowo omawiać.

Oglądając rysunek 3, łatwo zrozumieć, dlaczego przy tak wielu i tak bardzo różnych źródłach inspiracji – powstające metody sztucznej inteligencji nie miały ze sobą nic wspólnego i wytworzyły odrębne izolowane wyspy na wielkim oceanie niewiedzy (rys. 1).

Wiedząc o złożonym charakterze sztucznej inteligencji, spróbujmy ją jednak na początku ogólnie scharakteryzować.

Uwagi ogólne

Sztuczna inteligencja jest bez wątpienia częścią informatyki, chociaż ogromna ilość programów komputerowych ze sztucznej inteligencji nie korzysta, zaś niektóre aspekty sztucznej inteligencji, na przykład badana przez psychologów i filozofów kognitywistyka, nie mają bezpośredniego związku z informatyką. Obszary informatyki i sztucznej inteligencji pokrywają się więc ze sobą w znacznym stopniu, ale mają też obszary rozłączne (rys. 4).



Rys. 4. Relacje informatyki i sztucznej inteligencji

Istnieje wiele definicji sztucznej inteligencji, najkrócej jednak można powiedzieć, że ze sztuczną inteligencją mamy do czynienia wtedy, gdy maszyna (komputer albo elektronicznie sterowane urządzenie: robot, autonomiczny pojazd, samoorganizująca się sieć połączeń) przejawia zachowania, które obserwowane u człowieka powodowałyby, że bylibyśmy skłonni je uznać za skutek jego inteligencji.

Oczywiście ta definicja jest mało precyzyjna i w pewnych okolicznościach nieefektywna, ponieważ do oceny inteligencji ludzi i maszyn używamy różnych kryteriów. Na przykład jesteśmy skłonni uznać za inteligentnego ucznia, który doskonale przyswaja wiedzę i odpytywany przez nauczyciela potrafi tę wiedzę sprawnie przytoczyć. A tymczasem sieci Internet, w której zgromadzono zasoby wiedzy bez porównania większe, niż mógłby przyswoić sobie jakikolwiek człowiek, i która odtwarza zawartą w niej wiedzę ze stuprocentową dokładnością – za inteligentną nie uważamy. Owszem, inteligentne są programy, które tę sieć przeszukują i dostarczają nam na życzenie potrzebnych informacji – ale to jest osobna kwestia. Również używana do niedawna terminologia, wyróżniająca nazwą „inteligencja pracująca” osoby wykonujące niektóre prace (na przykład w księgowości) – w sytuacji, gdy rachunkowość współczesnych przedsiębiorstw prowadzi z reguły samodzielnie komputery, nie skłania nas do uznania ich „z automatu” za inteligentne.

Czy sztuczna inteligencja jest możliwa?

Inżynierowie mówią o sztucznej inteligencji na ogół rzeczowo i bez emocji. Jest ona składnikiem urządzeń, które budują albo które wykorzystują, więc zwykle nie stawiają pytania, czy owa sztuczna inteligencja jest możliwa, tylko jak jej skutecznie użyć. Ale humaniści z wytrwałością godną lepszej sprawy podnoszą kwestie semantyczne. Desygnatem rzeczownika „inteligencja” jest pewna cecha umysłu – ludzkiego lub niekiedy także zwierzęcego. Natomiast przymiotnik „sztuczna” określa wytwór techniki. Zatem „sztuczna inteligencja” to oksymoron – nazwa czegoś, co niejako z definicji istnieć nie może.

Dla przecięcia tych wątpliwości został zaproponowany tak zwany Test Turinga, przy pomocy którego ma dochodzić do rozstrzygnięcia, czy maszyna jest inteligentna, czy nie. Jego istotę przypomina rysunek 5.



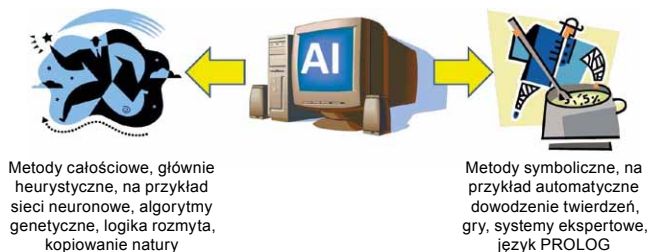
Rys. 5. Istota testu Turinga

W założeniach tego testu przyjmuje się, że arbiter (a w praktyce cała komisja arbitrów) komunikuje się z różnymi „rozmówcami” poprzez łącze nieujawniające ich tożsamości, ale pozwalające na wysyłanie dowolnych zapytań i na ocenę odpowiedzi (na przykład czat). Jeśli po dłuższej „rozmowie” ekspert nie potrafi odróżnić konwersującego z nim programu komputerowego od ludzi, z którymi równocześnie rozmawiał – to (zdaniem Turinga) ów program komputerowy jest inteligentny.

Dotychczas żaden z programów konwersacyjnych (tak zwanych chatbotów) nie „wygrał” testu Turinga, ale też we współczesnej sztucznej inteligencji niekoniecznie głównie chodzi o rozmowy z człowiekiem. Systemy sztucznej inteligencji są bowiem niekiedy używane jako inteligentna obsługa różnego rodzaju *call centers*, gdzie udzielają informacji i przyjmują zgłoszenia od rozmówców łączących się za pomocą e-mail albo telefonów, są w systemach ekspertowych (rys. 28), ale ponadto lokowane wszędzie: w fabrykach, w biurach, w gabinetach lekarskich, w pojazdach, w domach, we wszechobecnych smartfonach... Trudno wręcz wskazać miejsce, gdzie by stosowane nie były. Dlatego przyda się odrobina wiedzy na temat tego, czym jest owa sztuczna inteligencja, bo z reguły mówiące i piszące o niej osoby „wsadzają do jednego worka” bardzo różne rzeczy.

Najpierw ogólny podział

Wyruszając w „podróż” po archipelagu sztucznej inteligencji, dobrze jest mieć jego mapę. Zaczniemy jednak od zdefiniowania najbardziej ogólnego podziału. Generalnie sztuczna inteligencja jest rozwijana na dwa sposoby (rys. 6).



Rys. 6. Najbardziej ogólny podział metod sztucznej inteligencji

Po prawej stronie (na rysunku) są tak zwane metody symboliczne. Analizowany problem jest opisywany i zapisywany przy użyciu symboli, a algorytmy sztucznej inteligencji tak „mieszają” w tym zbiorze symboli, że w efekcie znajdowane jest rozwiązanie – też w postaci symbolicznej. To jest tradycyjna, najwcześniejsza rozwinięta, ale funkcjonująca do dzisiaj część sztucznej inteligencji. Będzie ona dalej omawiana. Przykładem zadań, które jako pierwsze udało się z sukcesem rozwiązać korzystając z metod symbolicznych, były programy automatycznie dowodzące twierdzenia matematyczne oraz grające w różne gry (warcaby, szachy, obecnie także w go).

Specjalnym działem metod symbolicznych są tak zwane systemy ekspertowe. Są to programy, które wykorzystując symboliczne metody automatycznego wnioskowania, odpowiadają na pytania użytkowników w podobny sposób, jak mógłby to czynić ludzki ekspert, czyli człowiek posiadający gruntowną wiedzę i mądrość niezbędną do tego, by udzielać mądrych rad. Systemy te też są opisane w jednym z dalszych rozdziałów.

Przy stosowaniu metod symbolicznych klasyczne metody programowania (na przykład popularny obecnie język C++) są mało użyteczne, został więc opracowany specjalny język do komputerowego manipulowania symbolami, o nazwie **LISP**. Ponadto komputery przetwarzające informacje w postaci symbolicznej często korzystają z logiki, dlatego powstał nawet specjalny język programowania o nazwie **PROLOG** (od *PROgraming in LOGic*, czyli programowanie w logice).

Metody oparte na rachunku symbolicznym są lubiane przez teoretyków, ponieważ na ich temat można sformułować i udowodnić mnóstwo ogólnych lematów i twierdzeń.

Jednak metodami opartymi na wykorzystaniu logiki i operacji na symbolach nie udaje się osiągnąć wszystkich celów stawianych przed sztuczną inteligencją, dlatego konieczne okazało się też alternatywne podejście. Polega ono na użyciu **metod całosciowych**, opartych na algorytmach heurystycznych. Termin **heurystyka** jest często używany w sztucznej inteligencji dla określenia metod, które – w odróżnieniu od dokładnych algorytmów – polegają na poszukiwaniu rozwiązania metodą swoistego „zgadywania”. Heurystyki działają bez gwarancji, że uda się znaleźć rozwiązanie, a gdy się je znajdzie, to czy będzie to rozwiązanie optymalne. Ale stosuje się je, bo są bardzo sprawne obliczeniowo, więc dla bardzo trudnych problemów (a takie z reguły pojawiają się w sztucznej inteligencji) można znaleźć rozwiązanie w akceptowalnym czasie. Dokładny algorytm nawet na najszybszym komputerze potrzebuje na przykład stu lat obliczeń, a metoda heurystyczna udziela odpowiedzi po kilku sekundach.

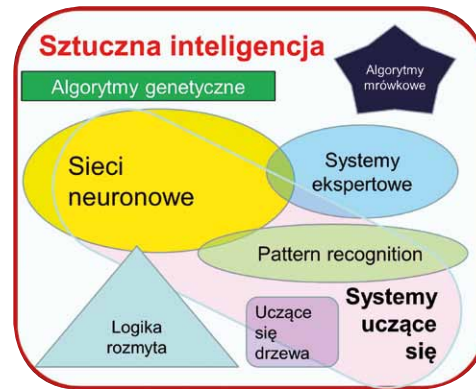
Istnieją różne metody heurystyczne, ale najczęściej stosowane są metody **uczenia maszynowego**, w których potrzebne rozwiązanie tworzone jest na zasadzie iteracyjnego doskonalenia pewnego początkowego (niedoskonałego) rozwiązania. To początkowe rozwiązanie podlega procesowi stopniowego doskonalenia aż do uzyskania rozwiązania zadowalającego pod względem jakościowym. Należą do tych metod w pierwszej kolejności bardzo popularne sieci neuronowe (będą dalej omawiane), ale używane są też uczące się drzewa decyzyjne, a nawet całe uczące się lasy. Warto dodać, że najstarszymi metodami opartymi na zasadach uczenia maszynowego były techniki rozwijane w ramach *pattern recognition*. Przytoczono tu (wyjątkowo!) angielską nazwę, bo jej polski odpowiednik jest mylący. Po polsku ten zbiór metod i algorytmów nazywany jest bowiem „rozpoznawanie obrazów”, co jest o tyle nietrafne, że owe metody są chętnie stosowane do klasyfikacji i kategoryzacji nie tylko obrazów jako takich (choć jest to dość częste ich zastosowanie [2]), ale również dźwięków (na przykład mowy [3]), symptomów pacjentów (w diagnostyce medycznej [4]), danych geofizycznych (przy poszukiwaniu złóż surowców, ale także ukrytych złók [5]), danych ekonomicznych (na przykład przy ustalaniu wiarygodności kredytobiorców [6]) i w mnóstwie innych kontekstów. Metody te są – jak wspomniano – najstarsze (pierwsza w Polsce była chyba para książek [7] i [8] wydanych w 1985 roku, naświetlająca te zagadnienia od strony teoretycznej i praktycznej), ale są ciągle rozwijane i doskonalone [9].

Do metod całosciowych zalicza się także algorytmy genetyczne, w których potrzebne rozwiązanie znajdowane jest w wyniku swoistej „hodowli” rozwiązań, z generowaniem

kolejnych „pokoleń” algorytmów. Doskonalenie działania następuje na zasadzie premiowania możliwością posiadania „potomstwa” tych algorytmów, które uzyskują najlepsze wyniki w testach. O algorytmach genetycznych też będzie dalej mowa, podobnie jak o innych metodach sztucznej inteligencji opartych na kopiowaniu natury (na przykład algorytmy mrówkowe). Do metod całościowych zaliczana jest także logika rozmyta i teoria zbiorów rozmytych – także dalej opisywana. O ile jednak sieci neuronowe naśladują fragmenty mózgu, a algorytmy genetyczne ewolucję, o tyle metody rozmyte modelują ludzki proces myślenia – mniej precyzyjny od komputerowego, ale często właśnie dlatego bardziej skuteczny.

Mapa klasyfikacji różnych metod sztucznej inteligencji

Na rysunku 7 przedstawiono swoistą „mapę” sztucznej inteligencji, na której zaznaczono proponowaną klasyfikację wymienionych wyżej metod, pokazując rozmiarami „obszarów” stopień wykorzystania tych metod, a wzajemnym (częściowym) przekrywaniem się obszarów – związki, jakie łączą te metody. Można więc na przykład zobaczyć, że do obszernej kategorii systemów uczących się należą między innymi sieci neuronowe. Ale widać też, że niektóre typy tych sieci nie zaliczają się do systemów uczących (żółty owal wystaje poza obręb zaokrąglonego różowego prostokąta). Bliższa analiza struktury



Rys. 7. Mapa pokazująca różne sposoby klasyfikacji metod sztucznej inteligencji

„mapy” pokazanej na rysunku 7 pozwala na wyciąganie licznych trafnych wniosków. Na przykład można zauważyć, że większa część problematyki systemów ekspertowych nie ma związku z uczeniem maszynowym, ale są też systemy ekspertowe wykorzystujące techniki uczenia się, z których niektóre angażują sieci neuronowe – ale nie wszystkie itd.

Mapa pokazana na rysunku 7 jest zbudowana na podstawie analizy **właściwości** różnych metod sztucznej inteligencji, które pod pewnymi względami są podobne, a pod innymi mogą się znacznie różnić. Ale bliskość czy nawet nakładanie się pewnych obszarów na przedstawionej mapie nie zmienia faktu, że bezpośrednie przejście od jednych metod do innych – nawet zaliczonych do tej samej kategorii – jest niemożliwe, czyli metafora

archipelagu (rys. 1) pozostaje aktualna. Obecnie rozpoczniemy wędrówkę po wybranych wyspach tego archipelagu.

Rachunek symboliczny jako najstarsza wyspa archipelagu

Spostrzeżenie, że komputery – początkowo traktowane wyłącznie jako maszyny liczące (nazwa „komputer” oznacza właśnie maszynę obliczającą) – mogą także manipulować symbolami, było przełomowym momentem. Gdy wykazano, że maszyny mogą operować na abstrakcyjnych pojęciach oznaczanych symbolami tak samo, jak na konkretnych liczbach, trzeba było uznać, że komputer to nie tylko szybkie liczydło. Ponieważ trzeba było jakoś owe symboliczne obliczenia wyodrębnić, więc dla tej rodzącej się dopiero dyscypliny maszynowego operowania na abstrakcjach zaproponowano nazwę „sztuczna inteligencja”. Wiadomo, gdzie i kiedy to było (podczas konferencji w Dartmouth College w 1956 roku), natomiast do dziś trwają spory, kto był autorem tej nazwy. Wielu uważa, że z propozycją wystąpił John McCarthy, chociaż inny z „ojców założycieli” sztucznej inteligencji, Marvin Minsky, uparcie twierdził, że to on był pomysłodawcą owej nazwy. Rozstrzygnąć się tego nie da, natomiast sam termin się przyjął i stał się na wiele lat hasłem, pod którym rozwijano najbardziej awangardowe działy informatyki.

Warto podkreślić, że przejście w technice komputerowej od obliczeń numerycznych do manipulacji symbolicznych miało naprawdę fundamentalne znaczenie. Obliczenia numeryczne mają to do siebie, że wykonuje się określone działania na konkretnych liczbach, uzyskuje się jakąś konkretną wartość wynikową – i nic więcej z tego nie wynika. Patrząc na zapis numeryczny:

$$2 \times 3 = 6$$

możemy skorzystać z wyniku (na przykład płacąc rachunek za 2 kg ziemniaków po 3 zł), natomiast trudno tu o jakiegłębniejszy wniosek.

Natomiast patrząc na zapis symboliczny:

$$m \times a = F$$

widzimy jedno z fundamentalnych praw fizyki, pozwalające na wyciąganie wielkiej liczby różnych wniosków. Na przykład można stwierdzić, jaką siłę (F) powinien rozwijać silnik samochodu o masie (m), żeby ten samochód uzyskał potrzebne przyspieszenie (a). Albo można stwierdzić, jaką masę (m) miał miecz kata ścinającego głowę skazańca (do czego potrzebna jest znana siła (F)), skoro przyspieszenie (a), jakie może rozwinąć ludzka ręka, jest ograniczone do znanej wartości. To dzięki znajomości tej formuły ludzie dotarli na Księżyc i zrozumieli tętno własnych serc.

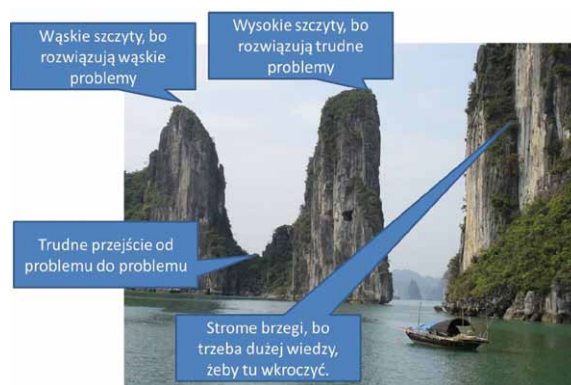
Przykładów można by było mnożyć bez liku, bo podane prawo fizyki opisuje **ruch dowolnego obiektu w dowolnych okolicznościach**. A jest takie uniwersalne, bo zamiast konkretnych liczb – występują w nim abstrakcyjne **symbole**.

Możliwość operowania przez komputer symbolami otworzyła drogę do maszynowego przekształcania formuł algebraicznych,

mechanicznego wyprowadzania potrzebnych wzorów matematycznych, a nawet do automatycznego dowodzenia twierdzeń matematycznych.

Symbole przekształcane przez komputer mogą reprezentować nie tylko abstrakcyjne pojęcia, ale również obiekty realnego świata. W ten sposób można komputerowi stawiać zadania, których rozwiązywanie powoduje określone działania w tym realnym świecie. Dzięki temu programy sztucznej inteligencji zaczęły grać w różne gry, rozwiązywać różne łamigłówki, a także operować na tekstach języka naturalnego. W ramach rozwoju metod symbolicznych powstawały na przykład pierwsze próby automatycznego tłumaczenia tekstów z jednego języka na inny. Wspomniany wyżej język **LISP** zaprojektował John McCarthy dla przetwarzania informacji symbolicznych – i to był ogromnie ważny fakt w dziejach sztucznej inteligencji.

Język ten jednym zachwycił (budowane były nawet specjalne komputery, których architektura logiczna dostosowana była do wykonywania programów pisanych w języku LISP!), a innych oburzał. Zasłużony pionier informatyki, Edsger Dijkstra, pisał, że używanie języka LISP to „najbardziej wyrafinowany sposób niewłaściwego używania komputerów”. Ale z pewnością prace pierwszych twórców programów operujących na symbolach stanowiły przełom w myśleniu o komputerach i obliczeniach.



Rys. 8. Metaforyczne przedstawienie wybranych cech symbolicznych metod sztucznej inteligencji

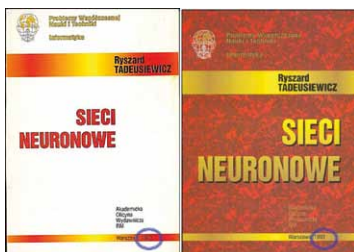
Wyspę, a raczej całą grupę blisko położonych wysp, wybranych do wyobrażenia metod symbolicznych, prezentuje rysunek 8. Takie są metody symboliczne – jeśli uzyskamy efektywne rozwiązanie dla jednego problemu (na przykład dowodzenia twierdzeń matematycznych), to nie bardzo możemy użyć tego rozwiązania jako bazy do – przykładowo – programu grającego w szachy. Same szczyty gór wznoszących się na wyspach są wysokie, bo problemy rozwiązywane tymi metodami bywają naprawdę trudne. Ale szczyty są wąskie, bo wąskie – w sensie merytorycznym – są rozwiązywane problemy. Na przykład program grający w szachy nie potrafi grać w warcaby. Na to, żeby przejść od programu skutecznie rozwiązującego jakąś jedną grupę problemów (na przykład rozwiązującego łamigłówki typu „kostka Rubika” poprzez zaplanowanie w określonej kolejności kolejnych ruchów) do problemu rozwiązującego inną grupę problemów (na przykład planującego działania robota, który ma wykonać zadanie wymagające wykonania w określonej kolejności kolejnych czynności), trzeba wykonać dużo pracy, bo oba programy są bardzo silnie związane z celami, którym służą.

Na domiar złego wyspa wyobrażająca metody symboliczne ma bardzo strome i wysokie brzegi. To metafora faktu, że na to, żeby rozpocząć pracę z którąś z metod symbolicznej sztucznej inteligencji, trzeba najpierw zdobyć dużą specjalistyczną wiedzę. Na przykład chcąc programować w języku LISP, trzeba najpierw poznać rachunek lambda, sprawiający kłopoty nawet biegłym w matematyce informatykom, oraz poznać technikę tworzenia S-wyrażeń. Przy stosowaniu metod symbolicznych odwołujących się do logiki matematycznej niezbędne jest korzystanie z rachunku predykatów pierwszego rzędu i rezolucji (na przykład przy automatycznym dowodzeniu twierdzeń) albo z rachunku zdań.

Z wymienionych powodów symboliczne metody sztucznej inteligencji, początkowo stanowiące główny napęd jej rozwoju, straciły na znaczeniu i dzisiaj są zdecydowanie rzadziej wykorzystywane.

Sztuczne sieci neuronowe

Kolejna przywołana tu wyspa symbolizuje sieci neuronowe. Do tych metod sztucznej inteligencji autor tego opracowania ma szczególny stosunek, bo jako jeden z pierwszych w Polsce zapoczątkował ich użytkowanie, pisząc w 1993 roku pierwszą książkę na ten temat [10]. Co ciekawe: książka ta, wydana po raz pierwszy w nakładzie 5 tys. egzemplarzy, miała wystarczyć na trzy lata. Została wykupiona w dwa tygodnie! Wydawnictwo w tym samym roku wydrukowało drugie wydanie (rys. 9).



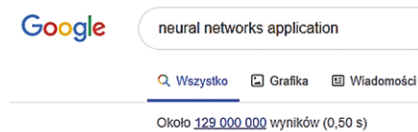
Rys. 9. Pierwsza polska książka na temat sztucznej inteligencji

Warto też odnotować fakt, że później (w 2007 roku) wydana została książka [11] opisująca, jak przy pomocy programów (udostępnionych za darmo w Internecie) można samemu w domu poznać właściwości sieci neuronowych, zarówno korzystając z programów udostępnionych w wersji wykonywalnej, jak i używając udostępnionych pełnych kodów źródłowych w języku C#, pozwalających na dodawanie aplikacji neuronowych do własnych programów rozwiązujących różne problemy. Książka [11] jest w całości dostępna w Internecie, a ponadto została przetłumaczona na język rosyjski i wydana w Moskwie, a także przetłumaczona na język angielski i wydana w USA (rys. 10). Programy udostępnione do książki były pobrane z serwisu internetowego ponad 66 tys. razy.



Rys. 10. Książka o sieciach neuronowych w polskiej, rosyjskiej i amerykańskiej wersji

To dzisiaj chyba najbardziej popularna metoda sztucznej inteligencji. Na zapytanie w Google w momencie pisania tego rozdziału (12.12.2020) *neural networks application* przyszła odpowiedź pokazana na rysunku 11.

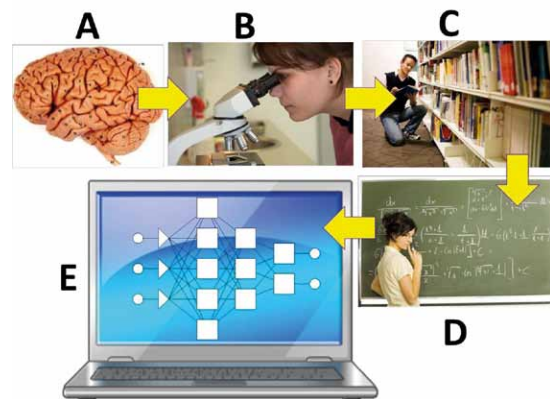


Rys. 11. Ilustracja popularności sieci neuronowych

Znaleziono na ten temat 129 milionów (!) dokumentów wyszukanych w ciągu pół sekundy. Dowodzi to, że sieci neuronowych jako narzędzi rozwiązywania różnych problemów z powodzeniem użyto ponad sto milionów razy. Wniosek, że użyto ich z powodzeniem, nasuwa się z takiego rozumowania: książkę, artykuł albo referat, które Google może odnaleźć w Internecie, pisze tylko badacz albo praktyk, który uzyskał jakiś sukces i go opisał. Jeśli komuś się coś nie uda – to nie pisze o tym artykule albo nie jedzie wypląkać się na konferencji, więc liczba tych negatywnych prób nie może być ustalona. Ale liczba sukcesów jest doprawdy imponująca!

Z kolei fakt, że Google dostarczył odpowiedzi w czasie pół sekundy (na podstawie zasobu ponad stu milionów dokumentów!) dowodzi, że analogiczne pytania musiało wcześniej stawić wielu Internautów i odpowiedź – uzyskana na podstawie przeszukania Sieci – musiała być już wcześniej przygotowana. A zatem liczba osób rozważających, czy użyć sieci neuronowych – musi być bardzo duża. To dobrze rokuje, jeśli idzie o dalszy rozwój tej metody!

Poznajmy je zatem nieco dokładniej: sieci neuronowe to narzędzie informatyczne, którego geneza przedstawiona jest skrótnie na rysunku 12.



Rys. 12. Sieci neuronowe powstały na podstawie badań mózgu

Punktem wyjścia było zainteresowanie mózgiem człowieka (A) – najbardziej złożonym i najbardziej tajemniczym ze wszystkich narządów. Ludzie od lat wiedzieli, że myślenie i inne procesy psychiczne zachodzą właśnie w mózgu, dlatego nie szczędzili wysiłku, żeby zbadać i opisać jego budowę i działanie. Trudziły się nad tym setki badaczy (B). Odkryto i zbadano bardzo wiele naukowych faktów na temat budowy i działania mózgu, tak że ostatnie 10 lat XX wieku nazwano „dekadą mózgu”. W efekcie prace neuroanatomów, neurobiologów

i neurofizjologów wypełniły całe biblioteki (C). Te zbiory opisanych faktów posłużyły biocybernetykom do tego, by stworzyć matematyczne modele funkcjonowania komórek nerwowych (neuronów) i ich zbiorowości – czyli sieci (D). Na podstawie tych modeli matematycznych stworzono programy symulujące funkcjonowanie sieci neuronowych (E). Początkowo te programy traktowano wyłącznie jako modele określonych biologicznych struktur nerwowych i rozwijano głównie ze względu na chęć lepszego zrozumienia działania mózgu. Potem jednak dostrzeżono, że te modele potrafią bardzo skutecznie rozwiązywać różne problemy, korzystając z dostępnych dla nich metod uczenia maszynowego. W efekcie przeniesiono – w pewnym zakresie oczywiście – intelekt człowieka, umiejscowiony w jego mózgu, do programu komputerowego (rys. 13).

Przykładowa struktura sieci neuronowej przedstawiona jest na rysunku 14. Zaznaczono tam warstwy neuronów wchodzących w skład sieci i podano ich role. Kwadratowe elementy oznaczają sztuczne **neurony**. Mogą to być specjalizowane urządzenia elektroniczne (buduje się obecnie takie neuronowe procesory) albo – częściej – fragment programu modelującego sieć neuronową na uniwersalnym komputerze. Sztuczne neurony naśladują niektóre funkcje rzeczywistych neuronów wchodzących w skład mózgu, ale funkcje te są maksymalnie uproszczone.

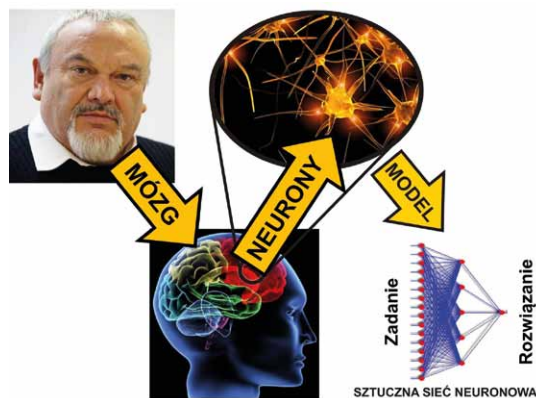
Ogólny schemat działania sieci neuronowej przedstawiono na rysunku 15.

Na rysunku tym neurony zaznaczono jako czerwone kropki. Niebieskie połączenia między neuronami odwzorowują łączące je **synapsy**. To właśnie zmiany w tych synapsach stanowią istotę procesu uczenia sieci, dlatego ich stan po procesie uczenia reprezentuje **wiedzę**, jaką sieć zdobyła w procesie uczenia. Jeśli do neuronów stanowiących **wejściową** warstwę podamy dane reprezentujące to zadanie, które aktualnie sieć ma rozwiązać, to owa wiedza pozwoli na takie działanie sieci, które doprowadzi do tego, że na **wyjściowej** warstwie sieci (na rysunku 15 złożonej tylko z jednego neurona, ale często mającej wiele neuronów, gdy rozwiązanie ma charakter wektorowy) pojawi się wynik, będący rozwiązaniem postawionego zadania.

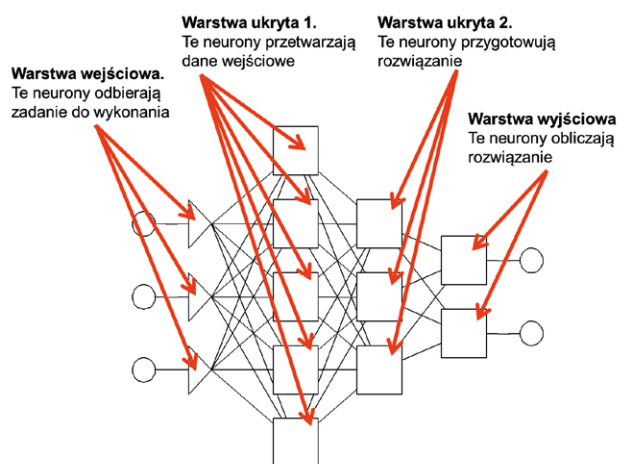
Charakteryzując sieci neuronowe zgodnie z przyjętą w tym artykule metaforą wysp, otrzymujemy ilustrację taką, jak na rysunku 16. Pierwsze, na co warto zwrócić uwagę na tym rysunku, to łagodne, przyjazne brzegi wyspy. Symbolizują one fakt, że początki korzystania z sieci neuronowych nie wymagają od użytkownika praktycznie żadnej wiedzy. Można zaczynać „z marszu”, zwłaszcza że istnieje wiele programów, zarówno płatnych, jak i darmowych, radykalnie ułatwiających tworzenie i użytkowanie tych sieci. O pewnym zbiorze takich programów była wyżej mowa w kontekście książki [11] i rysunku 10.

Ale to nie znaczy, że wiedza na temat sieci neuronowych jest nieprzydatna. Przeciwnie, im użytkownik lepiej pozna to narzędzie, tym lepsze efekty może uzyskać, co na rysunku symbolizuje rosnąca wysokość szczytów, które można osiągać, „zapuszczając się w głąb łądu”. Nie dało się tego pokazać na rysunku, ale jest faktem, że docelowo sieci neuronowe osiągną poziom przewyższający znacząco ten pułap, który były zdolne osiągnąć metody symboliczne!

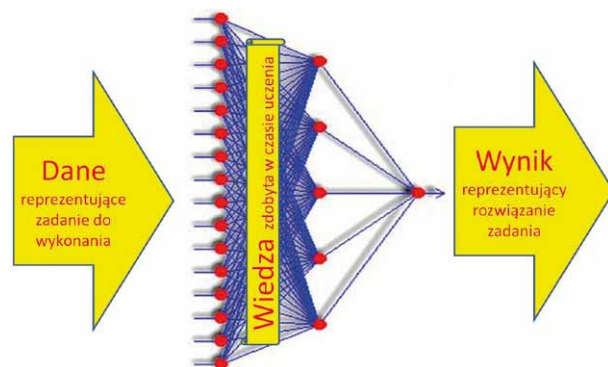
Oprócz typowych, najczęściej używanych sieci neuronowych, określanych często skrótem **MLP** (*Multi-Layer Perceptron*),



Rys. 13. Sieć neuronowa jako wynik modelowania fragmentów mózgu



Rys. 14. Struktura typowej sieci neuronowej



Rys. 15. Ogólny sposób działania sieci neuronowej

które reprezentuje główna część prezentowanej na rysunku 16 wyspy, zaznaczono na pierwszym planie dwie mniejsze wysepki, oddzielone od tej głównej, ale oparte na tej samej rafie. Są to rzadziej używane specjalne odmiany sieci neuronowych przeznaczone do specjalnych celów. Większa z tych wysepki reprezentuje tak zwane **sieci Kohonena**. Są to sieci **samouczące się**, które mogą prowadzić inteligentną analizę złożonych zbiorów danych bez udziału „nauczyciela”.

Kilka słów komentarza. O roli „nauczyciela” przy korzystaniu z sieci neuronowych (oraz innych metod opartych na uczeniu maszynowym) będzie mowa nieco dalej. Tu wystarczy tylko



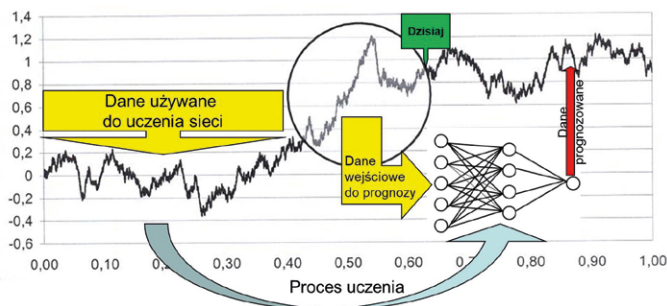
Rys. 16. Wyspa symbolizująca metody sztucznej inteligencji bazujące na sieciach neuronowych

odnotować, że typowe sieci neuronowe (w szczególności te najpopularniejsze, typu **MLP**) muszą być przed wykorzystaniem ich w praktyce uczone przez „nauczyciela”. Słowo „nauczyciel” trzeba tu traktować umownie – typowo jest to iteracyjny program komputerowy wyposażony w bazę wiedzy o tym, **co** należy rozwiązywać (w sensie takim, że pokazane są dane wejściowe oraz poprawne wyniki dla przykładowych zadań), ale nie ma żadnej informacji o tym, **jak** to należy rozwiązywać. Metodę rozwiązywania sieć musi stworzyć sama, opierając się na **indukcji**, czyli przejściu od szczegółowych przykładów do

ogólnej, *a priori* nieznannej, reguły. Warto dodać, że przeciwieństwem indukcji jest częściej stosowana **dedukcja**, polegająca na tym, że znając ogólną regułę, rozwiązuje się – korzystając z niej – wszelkie problemy szczegółowe.

Zdolna do indukcji sieć neuronowa to bardzo mocne i bardzo użyteczne narzędzie, szczególnie w zastosowaniu do zadań, dla których sami nie potrafimy zbudować algorytmu. Przykładem takich zadań są rozmaite prognozy, między innymi gospodarcze. Wiemy, że przyszłość zależy od stanu obecnego i od przeszłości, ale nikt nie ośmieli się podać równań pozwalających przewidzieć na przykład przyszłe wartości kursów walut. Natomiast sieć neuronowa może w trakcie nauki traktować sekwencje zdarzeń, które miały miejsce w przeszłości, jako przykłady działania tej nieznannej reguły, którą dopiero musi odkryć. Jest to możliwe, bo wiadomo, jakie zdarzenia zaszły wcześniej i jaki był ich skutek. Okazuje się, że sieć może na tej podstawie stworzyć model – na przykład rynku walutowego – i dość skutecznie przewidywać, co się wydarzy w przyszłości. Takich neuronowych prognoz używa się też w medycynie (przewidywanie skutków zastosowanej terapii) albo w meteorologii.

Na rysunku 16 przedstawiony jest schemat uczenia sieci neuronowej w zadaniu prognozowania. Omawiając krótko zasadę działania tej sieci, musimy zacząć od **danych**, które są używane do uczenia sieci. Dane te pozyskuje się z zarejestrowanej historii. Historia ta, w uproszczonym przykładzie zilustrowanym na



Rys. 17. Zasada uczenia sieci neuronowej wykorzystywanej do prognozowania

rysunku 17, ogranicza się wyłącznie do obserwacji tej jednej wartości, którą będziemy chcieli prognozować. Mogą to być na przykład wcześniejsze kursy tej waluty, której kurs następnego dnia chcemy odgadnąć. Ale warto dodać, że w rzeczywistych systemach prognostycznych do historii zalicza się (i dostarcza się do sieci neuronowej!) także inne ważne dane, które mogą wpływać na prognozę, na przykład kursy innych walut, wahania notowań giełdowych, ważne wydarzenia gospodarcze zachodzące poza sferą finansów, a nawet decyzje polityczne. Wszystkie te informacje, mogące mieć wpływ na prognozę, są rejestrowane w kolejnych dniach. Takie uporządkowane dane układają się w formę tak zwanego szeregu czasowego.

W takim szeregu można wydzielić (poczynając od dowolnego, losowo wybranego miejsca) zbiór wybranej liczby kolejnych obserwacji, traktowanych jako przesłanki do prognozy, oraz wartość obserwacji następującej zaraz po nich – jako tej, którą sieć powinna przewidzieć. Oczywiście ta prognozowana wartość jest znana, bo cały rozważany odcinek szeregu czasowego należy do przeszłości i wiadomo, co się zdarzyło, ale sieć tej prawdziwej wartości nie zna, tylko musi ją próbować odgadnąć. Takich „okienek” z przeszłości wybiera się dużo, bo danych do uczenia sieci musi być dużo. Potem jednak, gdy zasób danych przeznaczonych do uczenia zostanie wyczerpany, przechodzimy do punktu oznaczonego jako **dzisiaj**. Tu znamy tylko pewien fragment przeszłości (**dane wejściowe do prognozy**), a sieć ma przewidzieć, co się zdarzy w przyszłości (**dane prognozowane**).

O tym, jak przebiega proces uczenia i jak się wykorzystuje dane przeznaczone do uczenia sieci – będzie mowa w następnym rozdziale. Tu tylko pokazano zastosowanie tej ogólnej metody do szczególnego przypadku trenowania sieci mającej potem formułować prognozy.

Wracając do ogólnych rozważań na temat sieci neuronowych i do rysunku 16, trzeba podkreślić, że sieci neuronowe **Kohonena** mogą być użyte w tym przypadku, kiedy użytkownik nie potrafi zdefiniować celu analizy. Obrazowo można powiedzieć, że jest to narzędzie, które **potrafi udzielać odpowiedzi na nie zadane pytania**. Jeszcze inaczej można to wyrazić w następujący sposób: typowe sieci neuronowe mogą **zdobywać** wiedzę w trakcie procesu uczenia, ale musi istnieć źródło tej wiedzy (wspomniany wyżej „nauczyciel”). Natomiast sieci Kohonena same **odkrywają** wiedzę na podstawie analizy danych i mogą dostarczyć użytkownikowi zupełnie nowych i zaskakujących informacji.

Użycie sieci Kohonena wymaga pewnej wprawy, bo rozwiązania dostarczane przez nią mają formę tak zwanej mapy topologicznej, której interpretacja bywa trudna, ale wysiłek włożony w opanowanie tej sztuki sowicie się opłaca.

Na rysunku 15 widoczna jest jeszcze jedna mała wyseпка, związana (metaforycznie) z tak zwanymi **sieciami Hopfielda**. Są to sieci z bardzo silnymi mechanizmami rekurencyjnymi, mogące rozwiązywać problemy optymalizacyjne (znalezienie najlepszego rozwiązania przy obecności ograniczeń), a także mogą pracować jako tak zwane pamięci skojarzeniowe. Nie będziemy ich jednak tutaj dokładniej omawiać, bo są relatywnie rzadko stosowane.

No i na koniec wyłaniająca się na horyzoncie ogromna, ale jeszcze mgliście widoczna wyspa związana z tak zwanymi **sieciami głębokiego uczenia**. Odbiegają one od klasycznych sieci neuronowych, ponieważ mają bardzo wiele warstw neuronów (sieci **MLP** mają maksymalnie trzy) na przemian konwolucyjnych (*convolution layer*) i łączących (*pooling layer*) albo wykonujących podpróbkiowanie (*subsampling layer*). Używa się przy tym uczenia takich sieci metodami „bez nauczyciela” i „z nauczycielem”. Sieci te używane są od niedawna, ale liczba trudnych problemów, jakie zostały rozwiązane przy ich pomocy, jest już bardzo duża i stale rośnie – więc zdecydowanie warto się im uważnie przyglądać.

Uczenie maszynowe

Na chwilę odstawimy od rozważania kolejnych wysp naszego archipelagu i wyjaśnimy, jak przebiega proces uczenia maszynowego. Proces ten dotyczy wielu metod sztucznej inteligencji (przedstawiono to na rysunku 7), ale wygodnie będzie go przedstawić na przykładzie uczenia sieci neuronowej. Kluczem do tego uczenia jest tak zwany **zbiór uczący** – podstawowy motor napędowy wszystkich metod maszynowego uczenia. Zbiór uczący zawiera przykładowe zadania takiego typu, jakie sieć powinna potem rozwiązywać, przy czym w zbiorze tym zawarte są zarówno przesłanki (dane wejściowe), jak i wnioski (poprawne rozwiązania). Im większy zbiór uczący mamy do dyspozycji – tym lepszych wyników uczenia możemy oczekiwać. Ale ponieważ uczenie wymaga wielu kroków, więc nawet duży zbiór uczący szybko się wyczerpuje. Dlatego posiadany zbiór uczący wykorzystuje się w procesie uczenia wielokrotnie. Każdorazowe użycie wszystkich przykładów ze zbioru uczącego nazywa się **epoką**. Do uzyskania poprawnego działania sieci potrzeba często tysięcy lub nawet setek tysięcy epok!

Rozważmy przykładowy zbiór uczący (rys. 18). Zbiór ten jest przeznaczony do stworzenia sieci neuronowej odpowiadającej

MIASTO	TEMP	PRZEM	LUDN	FRED_W	OPAD	DNI_DESZ	SD2
Ploenza	70.3	213	582	6	7.05	36	10
Little R	61	91	132	8.2	48.52	100	13
San Fran	56.7	453	716	8.7	20.66	67	12
Denver	51.9	454	515	9	12.95	86	17
Hartford	49.1	412	158	9	43.37	127	56
Wilmington	54	80	80	9	40.25	114	36
Washington	57.3	434	757	9.3	38.89	111	29
Rockwood	68.4	136	529	8.8	54.47	116	14
Wien	75.5	207	335	9	59.6	128	10
Atlanta	61.5	368	497	9.1	48.34	115	24
Chicago	50.6	3344	3369	10.4	34.44	122	110

Rys. 18. Przykładowy zbiór uczący

na pytanie: Jak duże będzie zanieczyszczenie powietrza w różnych miastach? Oglądając przytoczoną tabelkę, widzimy kolejno od lewej kolumny. Pierwsza z nich nazwana jest MIASTO i podaje nazwę miasta, którego dane są dalej wymienione. Dana ta jest oczywiście nieistotna przy uczeniu sieci, dlatego widnieje jako zapisana słabą czcionką, ale jest ona potrzebna do tego, żeby sprawdzać poprawność danych i ewentualnie wprowadzać aktualizacje. W następnych kolumnach widzimy wartości, które będą potem danymi wejściowymi do sieci: TEMP – średnią roczną temperaturę (w stopniach Fahrenheita), PRZEM – liczbę zakładów przemysłowych, LUDN – liczbę ludności (w tysiącach), PRED_W – średnią prędkość wiatru, OPAD – wielkość średniorocznego opadu, DNI_DESZCZ – liczbę dni deszczowych w roku. Ostatnia kolumna, oznaczona SO₂, zawiera informacje, które powinny znaleźć się – jako rozwiązania – na wyjściu sieci. Jest to średnioroczne stężenie dwutlenku siarki w powietrzu, będące miarą jego zanieczyszczenia.

Samo uczenie przebiega według schematu pokazanego na rysunku 19. Obiektami uczestniczącymi w tym procesie są: sieć wymagająca uczenia, „nauczyciel” (pisany w cudzysłowie, bo w istocie jest to program komputerowy) oraz **zbiór uczący**. W przykładzie na rysunku 19 przyjęto, że w rozwiązywanym zadaniu występują trzy dane wejściowe i dwie dane wyjściowe, składające się na rozwiązanie zadania.

Te trzy obiekty wyróżniono na rysunku czerwonym kolorem. Pozostałe napisy na rysunku wiążą się z określonymi czynnościami i są zaznaczone na czarno. Omówimy teraz te czynności.

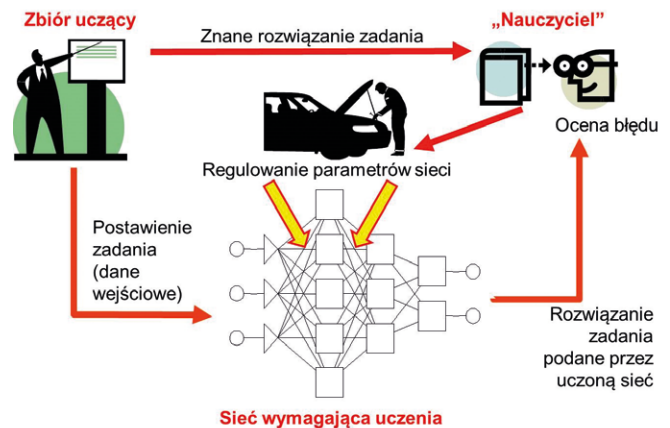
Każdy krok uczenia zaczyna się od **postawienia sieci zadania do wykonania**. Ze zbioru uczącego pobierane są dane wejściowe dla konkretnego przykładu i wprowadzane są na wejście sieci. W wyniku pracy sieci powstaje rozwiązanie zadania, które jest przedstawiane „nauczycielowi”. Ten ma możliwość skorzystania ze zbioru uczącego i może sprawdzić, jakie powinno być rozwiązanie zadania. Na tej podstawie powstaje ocena błędów popełnionych przez sieć, a gdy znamy błąd – to są znane metody **regulowania parametrów** sieci w taki sposób, by ten błąd zmniejszyć.

Jeśli konsekwentnie po każdym błędzie sieć jest dostrajana, to powinna działać coraz lepiej. Dążymy przy tym do tego, żeby sieć w końcu doszła do takiego stanu, by dla wszystkich zadań wchodzących w skład zbioru uczącego udzielała prawidłowych odpowiedzi. Nie zawsze to się udaje, a nawet jeśli się uda, to nie jest to jeszcze powód do uznania, że osiągnęliśmy końcowy sukces. Kolejnym problemem jest bowiem zdolność sieci do **uogólniania** zdobytej wiedzy.

Zapobieganie przeuczeniu

Fakt, że sieć prawidłowo rozwiązuje wszystkie zadania ze zbioru uczącego, jest mało przydatny w praktyce. Wszak dla tych zadań, które są w zbiorze uczącym, **znamy** rozwiązania, więc ich sprawne rozwiązywanie przez sieć neuronową w niczym nas nie wzbogaca. Te rozwiązania po prostu już mamy. Sieć będzie użytecznym narzędziem dopiero wtedy, kiedy potrafi skutecznie rozwiązywać zadania z nowymi, wcześniej niewidzianymi danymi.

W zadaniu prognozowania, od którego zaczęliśmy te rozważania, to przejście od zadań uczenia do zadań eksploatacji sieci



Rys. 19. Jeden krok procesu uczenia sieci. Objasnienia w tekście

wiąże się z konkretnym przekroczenia momentu czasowego oznaczonego na rysunku 17 jako „Dzisiaj”. Wszelkie prognozy dotyczące momentów czasowych przed „dzisiaj” mogą służyć jako elementy zbioru uczącego. Natomiast po przekroczeniu bariery „dzisiaj” zaczyna się poważna praca, która może mieć poważne konsekwencje. Sieć musi wtedy uogólniać wiedzę, jaką zdobyła w procesie uczenia, dokonując ekstrapolacji lub interpolacji posiadanych wiadomości. Tysiące przykładów zadań, skutecznie rozwiązywanym przez nauczone sieci neuronowe, są argumentem wskazującym, że takie uogólnianie wiedzy pochodzącej z procesu uczenia jest możliwe. Trzeba jednak zdawać sobie sprawę z tego, że to nigdy nie jest tak do końca pewne. Dlatego ostrożni użytkownicy metod sztucznej inteligencji opartych na paradygmacie uczenia maszynowego zwykle „nie odkrywają wszystkich kart” w trakcie uczenia.

Mając zgromadzony zbiór uczący, rozważny badacz dokonuje jego **losowego** (to bardzo ważne, że losowego!) podziału na trzy części. Większość przykładów uczących (70–80%) trafia do programu „nauczyciela” i jest motorem uczenia sieci (lub innej metody sztucznej inteligencji opartej na uczeniu). Ale część (zwykle około 20%) trzeba zachować jako tak zwane dane **walidacyjne**. Do czego one są potrzebne?

Otóż trzeba wziąć pod uwagę, że dane użyte do uczenia zawierają w sobie dwojakiego rodzaju informacje: takie, które wskazują, jak **generalnie** należy rozwiązywać zadania określonego typu, oraz takie, które związane są ze **szczegółową specyfiką** tych konkretnych przykładów, które zawarto w zbiorze uczącym. Na początku procesu uczenia sieć zdobywa głównie wiedzę ogólną. W efekcie coraz lepiej rozwiązuje zadania ze zbioru uczącego, ale coraz lepiej będzie sobie radziła także z nowymi, wcześniej niewidzianymi zadaniami.

Ale poczynając od pewnego momentu, kolejne korekty parametrów sieci zmuszają ją do tego, żeby coraz dokładniej odwzorowywała głównie te **specyficzne** informacje związane z indywidualnymi właściwościami tych zadań, których użyto do uczenia sieci. Pogarsza to jej użyteczność, bo w efekcie nowe zadania, z zasady odmienne od tych, które były zawarte w zbiorze uczącym, zaczynają być rozwiązywane coraz gorzej. Opisane zjawisko nazywa się **przeuczeniem** sieci.

Efekt ten można wykryć, stosując właśnie wspomniany wyżej zbiór walidacyjny. Jest to zbiór przykładów, dla których

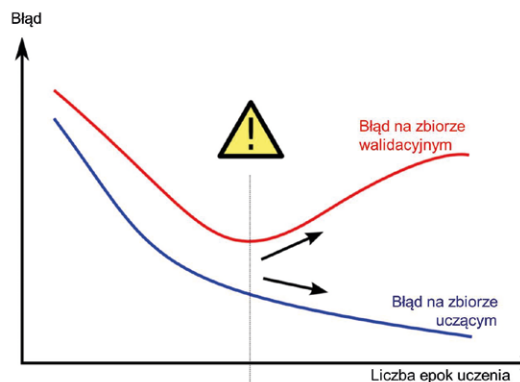
znamy poprawne rozwiązanie, ale których **nie używamy bezpośrednio do uczenia sieci**. Zbiór walidacyjny używa się po każdej epoce uczenia, obserwując, jak zmniejsza się błąd popełniany przez sieć w trakcie procesu uczenia dla danych, których do uczenia wprost nie używano. Nieco wyidealizowany wynik tej obserwacji przedstawiony został na rysunku 20. Widać, że początkowo błąd obserwowany na zbiorze uczącym maleje wraz ze wzrostem liczby epok uczenia (niebieska linia na rysunku), a błąd ustalany na zbiorze walidacyjnym maleje nieco wolniej (dlatego, że elementów zbioru walidacyjnego nie uwzględnia się przy uczeniu sieci), ale także spada. W pewnym momencie następuje jednak efekt „otwartych noży”: dalszemu maleniu błędu na zbiorze uczącym towarzyszy **wzrost** błędu określanego na zbiorze walidacyjnym. Należy wtedy bezwarunkowo przerwać uczenie, bo widać wyraźnie, że sieć traci zdolność do uogólniania wyników uczenia (rys. 20).

Przy okazji prezentacji krzywych uczenia na rysunku 20 warto może dodać jeszcze jedną informację dotyczącą w istocie większości procesów uczenia maszynowego. Otóż w praktyce proces uczenia nigdy nie przebiega tak gładko, jak by sugerowały książkowe wykresy. Prawdziwe procesy uczenia są obciążone licznymi czynnikami losowymi, związanymi z faktem, że poszczególne przykłady uczące mogą wywoływać takie poprawki parametrów sieci (lub innego algorytmu uczącego się), które polepszając ich działanie dla jednego przykładu, powodują pogorszenie dla innych przykładów, które powracając w kolejnych epokach, powodują chwilowe skokowe zwiększenie błędu. W efekcie prawdziwy przebieg błędu na zbiorze uczącym jest zwykle bardzo „nerwowy”, postrzępiony, co w oczywisty sposób przekłada się także na przebieg błędu na zbiorze walidacyjnym (rys. 21).

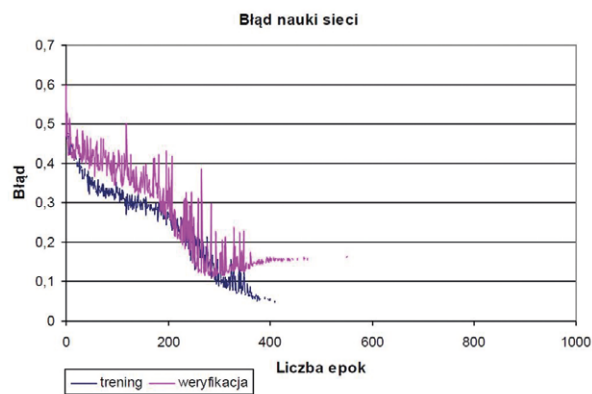
Utrudnia to wykrycie momentu, kiedy zaczyna się zjawisko przeuczenia. Teoretycznie powinno się przerwać uczenie, gdy obserwuje się wzrost błędu na zbiorze walidacyjnym przy równoczesnym maleniu błędu na zbiorze uczącym. Jeśli jednak oba błędy silnie oscylują, to trzeba stosować specjalne techniki (średnią kroczącą), żeby wykryć, kiedy **trend** malejący błędu w zbiorze walidacyjnym zamienia się w trend rosnący.

Jak wspomniano wyżej – zasób danych w zbiorze uczącym dzieli się (jeśli jego liczebność na to pozwala) na **trzy** części. Dwie poznaliśmy: to podzbiór danych przeznaczonych do uczenia i podzbiór służący do wykrycia momentu zatrzymania uczenia. A ten trzeci?

To zbiór **testowy**. Jeśli danych jest wystarczająco dużo, to około 10% przykładów badacz trzyma na boku i nie pozwala „nauczycielowi” na korzystanie z nich w trakcie uczenia ani do napędzania uczenia, ani do walidacji. Natomiast po zakończonym uczeniu sieć zostaje poddana „egzaminowi” – musi podać rozwiązania dla tych właśnie „odłożonych na bok” przykładów uczących. Jest to odpowiednik sytuacji, w jakiej sieć się znajdzie, gdy będzie musiała rozwiązywać nowe zadania, dla których nie będzie możliwości sprawdzenia, czy sieć się nie myli. Jeśli egzamin przeprowadzony z użyciem przykładów testowych przebiegnie pomyślnie (zwykle wykrywany błąd ma podobną wartość, jak na zbiorze walidacyjnym) – to można sieci zaufać przy rozwiązywaniu zadań praktycznych. Jeśli egzamin wypadnie niekorzystnie, to takiej sieci nie należy używać w praktyce.



Rys. 20. Wykrycie momentu początku przeuczenia przy użyciu danych walidacyjnych



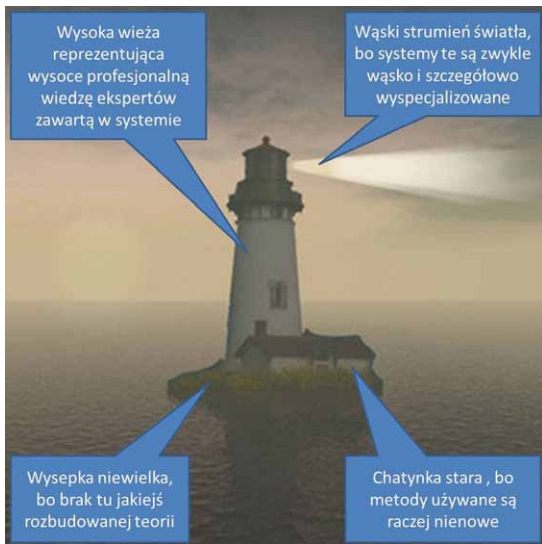
Rys. 21. Rzeczywisty przebieg błędu uczenia i błędu walidacji

Na rysunku 18 dane uczące są wpisane kolorem czarnym, dane walidacyjne – kolorem czerwonym, a dana testowa (jedna zaledwie w tym bardzo mało liczny zbiorze uczącym) – kolorem niebieskim. Oczywiście w tym przypadku jest to tylko ilustracja tego, jak to powinno wyglądać w prawdziwym zbiorze uczącym, liczącym zwykle przynajmniej kilkaset przykładów.

Światło ludzkiej mądrości w archipelagu sztucznej inteligencji

Sieci neuronowe wykorzystują osiągnięcia neurobiologii, która rozszyfrowała budowę i działanie naturalnego siedliska inteligencji, jakim jest ludzki mózg. Sztuczną inteligencję można jednak oprzeć także na innym źródle: na naśladowaniu procesów myślowych człowieka, badanych i odkrywanych przez psychologów. Przykładem systemów opartych na takich właśnie podstawach są tak zwane **systemy ekspertowe**. Są to programy, które potrafią gromadzić wiedzę ekspertów (ludzi), a potem ją wykorzystywać do tego, by doradzać użytkownikom (którzy ekspertami nie są), jak mają postępować przy rozwiązywaniu konkretnych problemów. Metafora tego działu sztucznej inteligencji w postaci umownej wyspy w opisywanym archipelagu przedstawiona jest na rysunku 22.

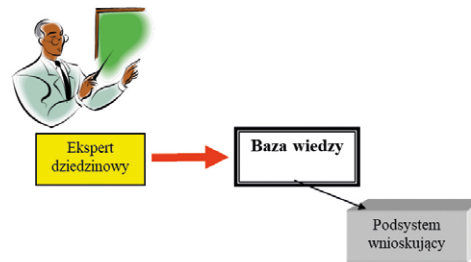
Ogólne właściwości systemów ekspertowych opisano na rysunku, więc teraz zobaczmy, jak taki system ekspertowy jest zbudowany i jak działa.



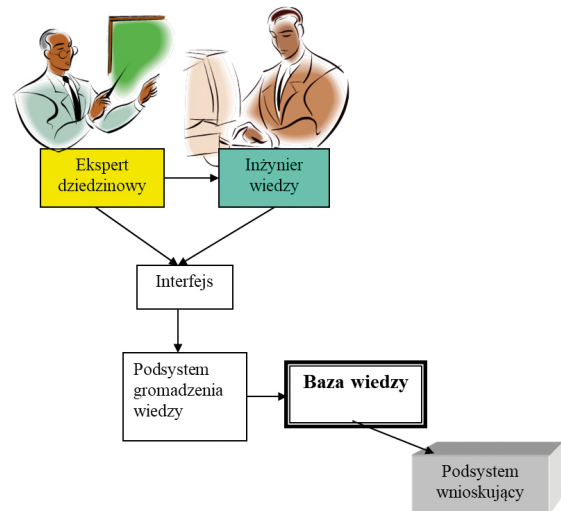
Rys. 22. Wyspa reprezentująca systemy ekspertowe

Działanie systemu ekspertowego, którego ogólny schemat przedstawiono na rysunku 22, składa się z dwóch etapów. Pierwszy polega na „karmieniu” systemu wiedzą. Potrzebna wiedza znajduje się w umysłach ekspertów z określonej dziedziny, na przykład lekarzy stawiających trafne diagnozy i osiągających sukcesy w leczeniu albo radców prawnych o dużym doświadczeniu. Gdy ta **wiedza** zostanie umieszczona w systemie ekspertowym w tak zwanej **bazie wiedzy** i zostanie połączona z wbudowanym **podsystemem wnioskującym** (rys. 23) – to użytkownicy będą mogli uzyskać odpowiedzi na swoje konkretne pytania albo rozwiązać nurtujące ich problemy.

Źródłem trudności podczas realizacji tego etapu jest fakt, że z reguły wybrany „ekspert dziedziny” nie potrafi swojej wiedzy przekazać komputerowi. On tę wiedzę posiada, ale nie w takiej formie, żeby mógł nią wprost zasilić **bazę wiedzy**, która jest centralnym elementem systemu ekspertowego. Na tym etapie ekspertowi pomagają informatycy nazywani **inżynierami**



Rys. 23. Zasilanie systemu ekspertowego wiedzą przez eksperta



Rys. 24. Pozyskiwanie wiedzy eksperta przez inżyniera wiedzy

wiedzy. Mają oni specjalne przeszkolenie psychologiczne, dzięki któremu umiejętnie przepytują ekspertów, a pozyskaną od nich wiedzę implementują w systemie komputerowym. Pomaga w tym specjalnie zaprojektowany interfejs eksperta oraz podsystem gromadzenia wiedzy (rys. 24).

Niebanalnym problemem jest to, jak pozyskaną wiedzę człowieka reprezentować w pamięci komputera. Jeśli wiedza dotyczy **faktów**, to jej zapis w komputerze jest łatwy. Umieemy budować

bazy danych, które zawierają miliony faktów traktowanych jako pewniki, więc to nie sprawia kłopotu. Ważniejsze i trudniejsze są jednak te fragmenty wiedzy eksperta, które dotyczą sposobów rozumowania. Można je odwzorować w komputerze na wiele sposobów, ale ogólnie jest to trudne. Najczęściej korzysta się z tak zwanych **metod regułowych**, to znaczy wyraża się wiedzę eksperta w formie reguły typu:

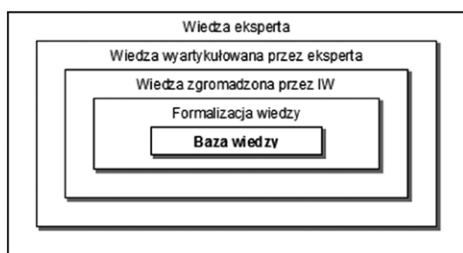
jeśli A to B

Oczywiście A i B są pieczołowicie i mądrze wybierane, ale jest to spora praca (dla inżyniera wiedzy), a ponadto trzeba dokładać starań, żeby uzyskane reguły w sposób wyczerpujący pokrywały całą wiedzę posiadaną przez eksperta. Co więcej, niekiedy reguły bywają sprzeczne (zwłaszcza gdy pozyskujemy wiedzę od wielu ekspertów) i wtedy trzeba te sprzeczności wyjaśniać. Ostatecznie uzyskujemy jednak bazę wiedzy, w której mieszczą się wszystkie pozyskane fakty i reguły (rys. 25).



Rys. 25. Schematyczna struktura bazy wiedzy

Warto zdawać sobie sprawę, że wypełnianie bazy wiedzy na podstawie wiedzy eksperta wiąże się ze stratami. Ekspert wiedzę posiada, ale nie w całości potrafi ją wyartykułować. Gdy już ją opisze – inżynier wiedzy nie wszystko zrozumie i nie wszystko zarejestruje. Gdy już wiedza jest zgromadzona, to trzeba ją sformalizować, żeby stała się dostępna dla komputera. No i wreszcie trzeba ją wbudować do bazy wiedzy, co także zwykle wiąże się z jejubożeniem (rys. 26).

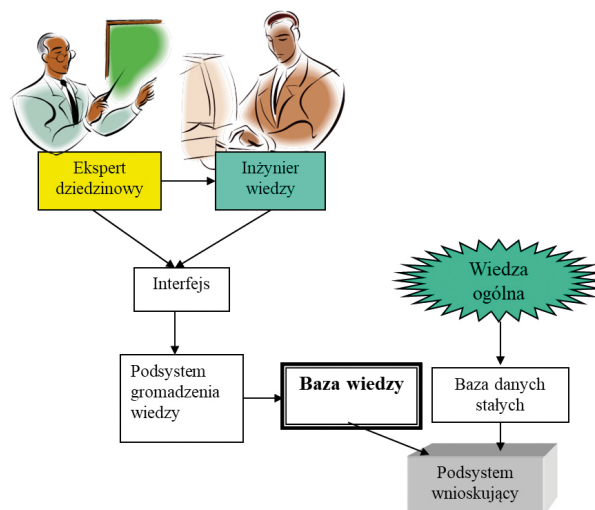


Rys. 26. Relacja między wiedzą eksperta i zawartością bazy wiedzy

Ale jeśli fakty i reguły dostarczone przez eksperta (nawet w uboższej formie) zostaną skompletowane i umieszczone w systemie – to pozwolą one na przejście do drugiego etapu, to znaczy do eksploatacji wypełnionego wiedzą systemu ekspertowego.

Zanim to jednak nastąpi – trzeba dodać jeszcze coś, z czego twórcy systemów sztucznej inteligencji początkowo nie zdawali sobie sprawy. Chodzi o tak zwaną **wiedzę ogólną**. Ekspert jej nie poda, bo pewne rzeczy są dla każdego człowieka oczywiste. A tymczasem dla komputera nic nie jest oczywiste, więc trzeba duże zasoby tej zdroworoządkowej wiedzy także do programu dostarczyć, żeby nie produkował paradoksów.

Nie wiadomo, na ile prawdziwa jest anegdota opowiadana w środowiskach informatycznych, ale doskonale ilustruje ona, na czym polega problem owej wiedzy ogólnej. Otóż podobno gdy zbudowano pierwsze systemy ekspertowe dla potrzeb medycyny (a warto wiedzieć, że to naprawdę jest obszar



Rys. 27. Struktura systemu uzupełnionego o bazę danych stałych

w którym powstało szczególnie wiele propozycji systemów ekspertowych), to lekarze chcieli poznać sposób zwalczania raka. W tym celu wyposażono system w wiedzę na temat biologii nowotworów i zapytano, co zrobić, żeby zniszczyć raka u konkretnego pacjenta. Odpowiedź komputera była szybka i logiczna: należy zabić tego pacjenta, a wtedy rak zginie. No bo żadna reguła podana przez ekspertów nie zabraniała zabijania pacjentów...

Anegdota może nie jest prawdziwa, ale problem jako taki jest prawdziwy: używając systemów ekspertowych (lub innych metod sztucznej inteligencji) jako narzędzi wspomagających proces podejmowania decyzji, musimy bardzo baczyć na to, by wyposażać je także w zasoby wiedzy ogólnej, bo sama wiedza specjalistyczna, nawet ta na najwyższym poziomie, na ogół nie wystarcza.

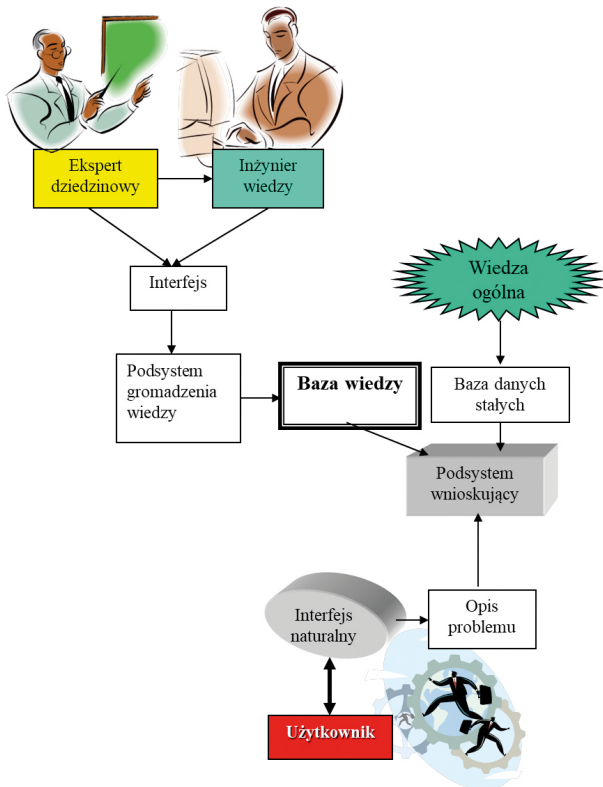
Dlatego w strukturze systemu ekspertowego musi pojawić się jeszcze jeden zasób informacji zawierający właśnie tę wiedzę ogólną, nazywany **bazą danych stałych** (rys. 27).

Eksploatacja systemu ekspertowego

Mając wypełnioną bazę wiedzy (specjalistycznej) oraz bazę danych stałych, możemy już przystąpić do drugiego etapu, to znaczy do eksploatacji systemu. Sens istnienia systemów ekspertowych polega na tym, że mogą one być wykorzystywane przez wielu użytkowników, którzy – jak to zasugerowano na rysunku 28 – zwykle bardzo się spieszą i chcą skorzystać z mądrości ekspertów zawartej w systemie jak najszybciej i jak najskuteczniej.

Użytkownicy systemu mogą z niego korzystać na różne sposoby. Najczęściej potrzebują rady, stąd systemy ekspertowe bywają często nazywane systemami doradczymi. Ale mogą też być inne potrzeby, o których wspomnimy na końcu tego rozdziału.

Użytkownik opisuje swój problem, a system automatycznie przeprowadza wnioskowanie, dla którego punktem startowym jest pytanie użytkownika. Dość ważnym elementem w tej części systemu, która ma styczność z użytkownikami, jest **interfejs naturalny**. Przymiotnik „naturalny” oznacza w tym przypadku, że użytkownicy swoje pytania mogą swobodnie formułować

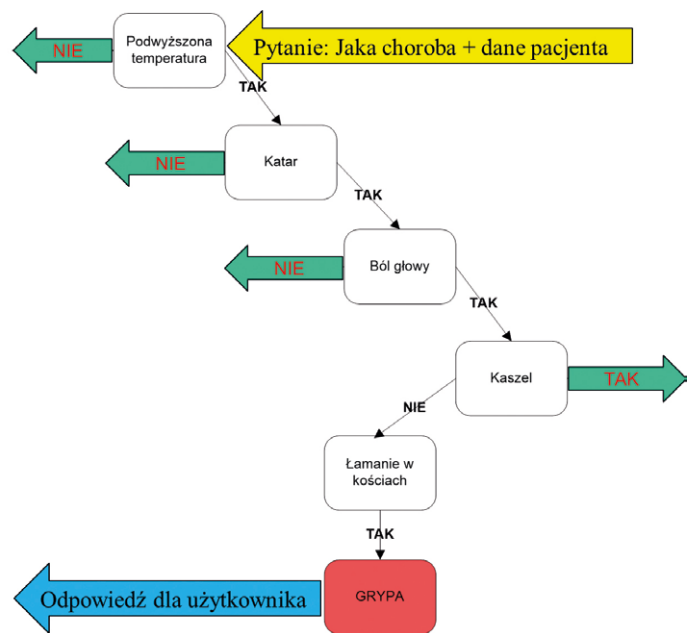


Rys. 28. Pozyskanie opisu problemu od użytkownika poprzez interfejs naturalny

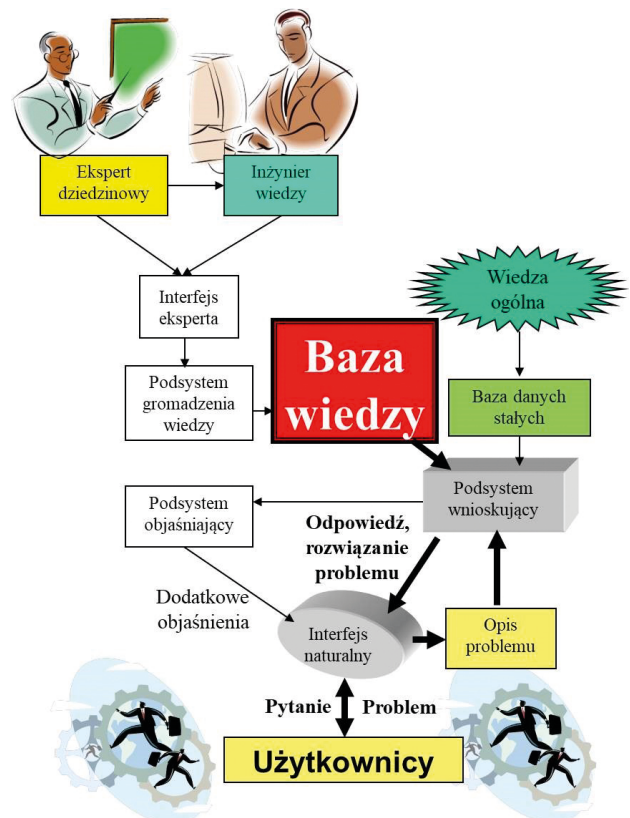
w języku naturalnym (na przykład po polsku), a system usiłuje z ich wypowiedzi wytworzyć odpowiednio zakodowany opis problemu. Czasem konieczne jest „dopytanie” użytkownika, czy właśnie o to mu chodzi, albo uzgodnienie w trybie dialogu potrzebnych szczegółów. W efekcie tej wymiany informacji między użytkownikiem a systemem powstaje **opis problemu** (rys. 28).

Opis problemu staje się początkiem drogi automatycznego rozumowania, realizowanego przez **podsystem wnioskujący**.

Ten element systemu ma oczywiście podstawowe znaczenie przy jego eksploatacji i był dość trudny do zbudowania, gdy tworzone go po raz pierwszy. Ale udało się, bo punktem wyjścia do konstrukcji podsystemów wnioskujących były stworzone jeszcze w latach 50. XX wieku pogromy automatycznego dowodzenia twierdzeń matematycznych. Okazało się, że droga rozumowania, jakie trzeba przeprowadzić, aby dla problemu opisanego przez użytkownika znaleźć potrzebne rozwiązanie przy wykorzystaniu wiedzy ekspertów – jest bardzo podobna do tej drogi, jaką przebywa komputer, dążąc do udowodnienia tezy matematycznego twierdzenia przy wykorzystaniu sformułowanych założeń i istniejącej wiedzy matematycznej. W jednym i w drugim przypadku „drogowskazem” jest metoda rezolucji Robinsona. To dzięki tej metodzie tak są automatycznie dobierane fakty i reguły zawarte w bazie wiedzy, aby stworzyć łańcuch przesłanek i częściowych wniosków, a na końcu, jako ostatni wniosek, uzyskać odpowiedź adresowaną do użytkownika, będącą rozwiązaniem jego problemu. Bardzo uproszczony schemat automatycznego wnioskowania pokazany jest na rysunku 29.



Rys. 29. Przykładowy przebieg automatycznego wnioskowania



Rys. 30. Pełny schemat systemu ekspertowego

Oczywiście przy praktycznej realizacji opisanej koncepcji trzeba było rozwiązać mnóstwo problemów szczegółowych, ale obecnie jest to na tyle dobrze dopracowane i na tyle uniwersalne, że sprzedawana są systemy ekspertowe „szkieletowe”

zwierające gotowy do użycia podsystem wnioskujący, a także interfejs naturalny dla użytkownika i interfejs eksperta – ale z całkowicie pustą bazą wiedzy. Użytkownik może wypełnić system wiedzą na dowolny wybrany przez siebie temat i mieć w ten sposób elektronicznego eksperta znającego dokładnie jego unikatowe potrzeby i jego uwarunkowania.

Efektom działania podsystemu wnioskującego w systemie wypełnionym adekwatną wiedzą jest **odpowiedź**, jaką system przekazuje użytkownikowi za pośrednictwem interfejsu naturalnego (rys. 30).

Odpowiedź ta może stanowić rozwiązanie problemu nurtującego użytkownika albo może być wiadomością, że przy wszystkich założeniach, które użytkownik umieścił w opisie problemu – rozwiązania nie da się znaleźć. W tym drugim przypadku użytkownik jest zachęcany, żeby dostarczył więcej danych albo zrezygnował z części wymagań. Zwykle po kilku uściśleniach rozwiązanie problemu zostaje znalezione i użytkownik może z niego skorzystać. Ale może się zdarzyć użytkownik szczególnie wymagający, który nie tylko wymaga rozwiązania problemu, ale dodatkowo może się domagać **uzasadnienia**, dlaczego to rozwiązanie jest właśnie takie, a nie inne. System ekspertowy jest na to przygotowany i specjalnie zaprogramowany jego fragment, **nazywany podsystemem objaśniającym**, przekazuje poprzez interfejs naturalny całą drogę rozumowania, jaka doprowadziła do podanego rozwiązania. Widać, jak wychodząc od opisu problemu podanego przez użytkownika i korzystając z kolejnych reguł umieszczonych przez ekspertów w bazie wiedzy osiąga się rozwiązanie, podane użytkownikowi jako rozwiązanie. Użytkownik może przyjąć te wyjaśnienia albo może tak zmienić swój opis problemu, żeby ukierunkować automatyczne rozumowanie na ścieżkę bardziej zgodną z jego oczekiwaniami. Może także wskazać elementy bazy wiedzy (fakty albo reguły), których do tego jego problemu stosować nie należy. Proces automatycznego rozumowania zostanie wtedy zainicjowany ponownie i dostarczy być może innego rozwiązania, które będzie poprawne i logiczne w świetle nowego opisu problemu oraz na chwilę (tylko dla tego problemu) zmodyfikowanej bazy wiedzy.


Na koniec tego rozdziału warto wspomnieć o jeszcze jednej ciekawostce związanej z systemami ekspertowymi. Otóż większość z nich działa w ten sposób, że **doradza** użytkownikom, stąd dość popularna jest także nazwa „systemy doradcze”. Ale jest też odmiana, która **krytykuje** pomysły użytkownika i w ten sposób przyczynia się do ich doskonalenia. Wyobraźmy sobie, że prezes dużej firmy wymyślił nową strategię jej działania i zastanawia się, czy jest to dobra strategia. Nie może pokazać tego pomysłu swoim współpracownikom, bo nikt nie ośmieli się krytykować prezesa. Tym bardziej nie może tego pokazać nikomu z ludzi z zewnątrz, bo powodzenie zależy od zachowania tych planów w tajemnicy. Ale może pokazać ten plan systemowi ekspertowemu. On się nie zawaha i skrytykuje wszystko, co takiej krytyki wymaga, przeprowadzi symulację przewidywanych skutków, budując odpowiednie prognozy, wykryje i wskaże każdy słaby punkt. A na komputer prezes się przecież nie może obrazić, tylko poprawi swój plan i podda go kolejnej ocenie – i tak aż do skutku.

Uwagi końcowe

Temat archipelagu sztucznej inteligencji okazał się zbyt obszerny, by udało się go zawrzeć w jednym artykule. Stąd omówiwszy (i to w wielkim skrócie!) część podjętej problematyki, przeniesiemy dyskusję dalszych metod sztucznej inteligencji wzmiankowanych na rysunku 7 do kolejnego artykułu w następnym numerze miesięcznika „Napędy i Sterowanie”. Jeśli kogoś z Czytelników artykuł ten zainspirował do przemyśleń, czy to dobrze, że rozwijamy sztuczną inteligencję, czy też może wiąże się z tym jakieś zagrożenia – zachęcamy do lektury pozycji [12] spisu literatury.

Literatura

- [1] TADEUSIEWICZ R.: *Zdefiniujmy przedmiot debaty: czym jest oraz czym nie jest sztuczna inteligencja?* Rozdział w książce: *Ergonomia wobec idei sztucznej inteligencji*. Wydawnictwo Politechniki Krakowskiej, Kraków 2020.
- [2] TADEUSIEWICZ R.: *Systemy wizyjne robotów przemysłowych*. WNT, Warszawa 1992.
- [3] TADEUSIEWICZ R.: *Sygnal mowy* (Monografia książkowa wydana w serii *Problemy elektroniki i telekomunikacji*). WKiŁ, Warszawa 1988.
- [4] TADEUSIEWICZ R., ŚMIETAŃSKI J.: *Pozyskiwanie obrazów medycznych oraz ich przetwarzanie, analiza, automatyczne rozpoznawanie i diagnostyczna interpretacja*. Wydawnictwo STN, Kraków 2011.
- [5] MAZURKIEWICZ E., TOMECKA-SUCHOŃ S., TADEUSIEWICZ R.: *Application of Neural Network Enhanced Ground Penetrating Radar to Localization of Burial Sites*. „Applied Artificial Intelligence”, vol. 30, No. 9, 2016.
- [6] LULA P., PALIWODA-PĘKOSZ G., TADEUSIEWICZ R.: *Metody sztucznej inteligencji i ich zastosowania w ekonomii i zarządzaniu*. Wydawnictwa Akademii Ekonomicznej w Krakowie, Kraków 2007.
- [7] TADEUSIEWICZ R.: *Rozpoznawanie obrazów – zarys teorii*. Skrypt UJ nr 499, Kraków 1985.
- [8] TADEUSIEWICZ R.: *Rozpoznawanie obrazów w zastosowaniach ekonomicznych*. Academy of Economics Lecture Notes – Skrypt AE, Kraków 1985.
- [9] TADEUSIEWICZ R., FLASIŃSKI M.: *Rozpoznawanie obrazów*. PWN, Warszawa 1991.
- [10] TADEUSIEWICZ R.: *Neural Networks* [in Polish: *Sieci neuronowe*], Akademicka Oficyna Wydawnicza, Warszawa 1993.
- [11] TADEUSIEWICZ R., GĄCIARZ T., BOROWIK B., LEPEK B.: *Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#*. Wydawnictwo Polskiej Akademii Umiejętności, Kraków 2007.
- [12] TADEUSIEWICZ R.: *Automatyzacja i sztuczna inteligencja jako źródła prawdziwych i wyimaginowanych zagrożeń*. Rozdział w książce: *GALWAS B., KOZŁOWSKI P., PRANDECKI K. (RED.): Czy świat należy urządzić inaczej – schyłek i początek*. Komitet Prognoz „Polska 2000 Plus” przy Prezydium PAN, Warszawa 2019.

 prof. dr hab. inż. Ryszard Tadeusiewicz
AGH – Akademia Górniczo-Hutnicza w Krakowie
e-mail: rtad@agh.edu.pl, www.Tadeusiewicz.pl