X Konferencja PLOUG
Kościelisko
Październik 2004

Sieci neuronowe a energetyka – prawdy i mity o prognozowaniu

Tomasz Wyrozumski

Biuro Matematyki Stosowanej S.C.
e-mail: tw@bms.krakow.pl

Abstrakt
Artykuł poświęcony jest praktycznemu wykorzystaniu sztucznych sieci neuronowych. Wychodząc od podstaw omawiamy zasady działania perceptronu nieliniowego i sieci perceptronowych oraz jedną z najważniejszych metod ich uczenia. Koncentrujemy się głównie na praktycznych aspektach zastosowania prezentowanych układów do prognozowania, operując przy tym na konkretnym, rzeczywistym przykładzie przewidywania zużycia energii elektrycznej w systemie *Syndis-Energia*, wyposażonym w moduł *N-Predictor*. Wskazujemy na ewidentne zalety, ale też ograniczenia sieci jako narzędzi prognostycznych. Polemizując z pewnymi obiegowymi poglądami dotyczącymi konfigurowania, trenowania i zasilania danymi układów neuropodobnych, staramy się jednocześnie udzielić czytelnikowi szeregu praktycznych porad odnośnie ich skutecznego wykorzystywania.

1. Na czym polega prognozowanie?

Prognozowanie to przewidywanie przyszłości, a przyszłość niemal z definicji jest czymś nieprzewidywalnym. Tak w każdym razie postrzegają ją nasze umysły. Z tego względu prognozowanie wywołuje niekiedy skojarzenia ze szklaną kulą i oczywiście – ironiczne uśmiechy racjonalnie myślących ludzi. Zapewne nie wszyscy czytamy horoskopy i chodzimy do wróżek (autor chciałby wierzyć, że czyni to jedynie znikomy margines społeczeństwa), jednak większość z nas uważnie śledzi prognozy pogody i stosownie do nich dobiera ubiór, bądź też planuje takie, czy inne działania. Trzeba uczciwie przyznać, że jakość tych prognoz niezwykle wzrosła w ciągu ostatnich dziesięcioleci i choć bywa, że się one całkowicie nie sprawdzają, to jednak ogólnie trafność jest tu imponująca. Co zatem różni prognozy jasnowidza od prognoz meteorologa? To, że prognozy meteorologa są produktem pewnego modelu, zaś jeśli chodzi o prognozy jasnowidza – nie posiadamy niestety orientacji, skąd się one biorą.

Istota racjonalnego prognozowania polega zatem na budowaniu modeli rzeczywistości, a następnie obserwowaniu, jak się one zachowują z upływem (modelowego) czasu i dokonywaniu projekcji wyników z powrotem na rzeczywistość. Jakość prognozy zależy zasadniczo od dwóch czynników – od tego na ile model jest ścisły, tzn. sformalizowany matematycznie i na ile wiernie odzwierciedla realia. Dla przykładu, ruch kul bilardowych po stole da się opisać przy pomocy powszechnie znanych równań Newtona, które leżą u podstaw znakomicie sformalizowanego modelu. Prognozowanie, gdzie i kiedy znajdzie się uderzona z określoną siłą kula nie ma nic wspólnego z wróżeniem, a trafność prognozy jest praktycznie stuprocentowa. Świadomie dodaliśmy przy tym słowo „praktycznie", bynajmniej nie dlatego, że kwestionujemy zasady dynamiki, lecz dlatego, że rzeczywistość na ogół jest bardziej złożona niż model. Jeśli prognozując ruch kul założymy, że są one idealnie sprężyste, to na pewno dokonamy zaledwie pewnego przybliżenia rzeczywistości, przybliżenia, które nie pozostanie bez wpływu na jakość prognozy, podobnie zresztą jak i inne przybliżenia, np. założenie że współczynnik tarcia kul o sukno jest stały w każdym punkcie stołu, że płaszczyzna stołu jest idealnie pozioma, że opór powietrza można zaniedbać itd. Kto rozwiązywał kiedykolwiek zadania z fizyki, ma świadomość, że każde takie założenie jest bardzo przyjemne, ponieważ szalenie upraszcza rachunki. Jeśli dodatkowo pozostaje ono bez istotnego wpływu na jakość prognozy, to już na pewno warto je zrobić. Niestety, niekiedy bywa, że założenia upraszczające są nie do zaakceptowania. Wtedy na ogół czekają nas długie i żmudne obliczenia, jednak od czego w końcu są komputery? Nie powinniśmy narzekać – problem w tym, że może być znacznie gorzej. Nie zawsze bowiem model rzeczywistości, którym dysponujemy da się wywieść z jakichś praw fundamentalnych, takich jak wspomniana dynamika Newtona. Czasami musimy skonstruować go ad hoc, zastępując porządną teorię mniej lub bardziej naciąganą argumentacją, która przekłada się na matematyczne formuły. Z takim fenomenologicznym podejściem mamy do czynienia z reguły tam, gdzie stopień skomplikowania zjawiska jest bardzo duży, znacznie przewyższający zagadnienie bilardu. Niestety, bardzo często zdarza się to w problemach technicznych i ekonomicznych, ale też złożonych układach fizycznych, takich jak choćby jądra atomowe, gwiazdy, czy atmosfera, interesująca nas z uwagi na wspomniane już prognozowanie pogody.

Czy może być jeszcze gorzej? Niestety tak! Czasami po prostu w ogóle nie dysponujemy żadnym sensownym modelem, czyli krótko mówiąc mamy tylko mgliste pojęcie o istocie zjawiska. Na ogół uważa się, że w takiej sytuacji oprócz wróżki pozostaje jeszcze ekstrapolacja przebiegu czasowego. Istnieją różne warianty tego podejścia, jednak w gruncie rzeczy sprowadza się ono do dopasowania pewnej zależności funkcyjnej f do znanej części przebiegu, a następnie odczytania wartości funkcji dla przyszłych chwil czasu. Dopasowanie przeprowadza się metodą najmniejszych kwadratów, tzn. minimalizując wyrażenie

$$\sum_{i=1}^{n} [f(t_i) - x_i]^2 ,$$

gdzie x_i oznaczają rzeczywiste wartości, zmierzone w chwilach t_i. Minimum wyznaczamy ze względu na parametry funkcji (na przykład współczynniki wielomianu), ustalając tym samym jej

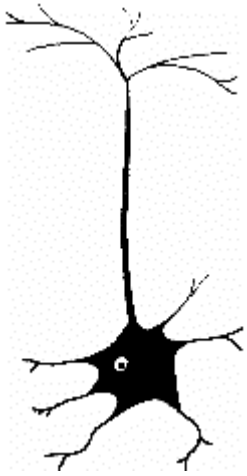
dokładną postać. W praktyce do zadanych punktów pomiarowych da się – lepiej lub gorzej – dopasować każdą funkcję, a to oznacza, że cała procedura wydaje się nieco absurdalna. Istotnie, aby miała ona jakikolwiek sens, potrzebne jest jeszcze jakieś aprioryczne założenie o wspomnianym przebiegu funkcyjnym.

Opisaną powyżej metodą standardowo posługują się fizycy, przy czym założeń zawsze dostarcza teoria, a dopasowanie zadanej krzywej do punktów pomiarowych pozwala wyznaczyć jej wolne parametry. Przykładowo, z dynamiki wiemy, że ześlizgujący się z równi pochyłej przedmiot porusza się ruchem jednostajnie przyspieszonym, a zatem mając pomiary odległości przebytych przezeń w kolejnych sekundach, powinniśmy wpasowywać w nie funkcję kwadratową. Dalszy przebieg funkcji będzie prognozą ruchu wspomnianego przedmiotu, a wyraz przy drugiej potędze czasu – przyspieszeniem. W eksperymencie wyznaczymy więc przyspieszenie, czyli pośrednio – współczynnik tarcia o równię (jeśli oczywiście przyspieszenie ziemskie już znamy). Ktoś, kto nie ma pojęcia o dynamice, mógłby jednak spróbować użyć wielomianu wyższego stopnia i przez przypadek uzyskać nawet w pewnym obszarze lepsze dopasowanie (pamiętajmy, że pomiary obarczone są błędem!), jednak im dłuższy horyzont czasowy, tym gorsze byłyby jego prognozy. I znów widzimy, że w tym, zdawałoby się nader fenomenologicznym podejściu musi pojawić się model, determinujący postać postulowanej zależności funkcyjnej.

Jeśli nie dysponujemy żadną, nawet najbardziej przybliżoną teorią, lepiej nie stosować tego typu procedur. Teorię należy więc albo stworzyć, albo zlecić jej stworzenie „komu innemu". Cudzysłów nie jest tu bynajmniej redakcyjną pomyłką – użyliśmy go świadomie, gdyż teorię może za nas stworzyć automat lub, jak kto woli, sztuczna inteligencja. Ponieważ taka wizja wydaje się dość kusząca (niebezpieczna?), jej praktycznej realizacji poświęcamy następne rozdziały.

2. Od neuronu do sieci neuronowej

Wyobraźmy sobie prosty układ przetwarzający informację, który posiada kilka wejść i jedno wyjście, tak jak to obrazuje Rys. 1.



Rys. 1. Komórka nerwowa

Jak wiadomo, z miliardów takich układów zbudowany jest nasz mózg. Przedstawiony poniżej sposób działania neuronu i konstrukcji na nim opartych są jedynie pewnym przybliżeniem skomplikowanej biologicznej rzeczywistości, jak się jednak okazuje, przybliżeniem na tyle dobrym, że symulujące owo działanie oprogramowanie komputerowe wykazuje pewne cechy istot myślących, zasługując tym samym na miano sztucznej inteligencji. Przyjrzyjmy się zatem dokładniej neuronowi. Ze względu na sposób przetwarzania sygnałów jest on elementem nieliniowym, tzn. zależność pomiędzy sygnałami x_i, podawanymi na jego kolejne wejścia (dendryty), a wartością na wyj-

ściu (neurycie lub inaczej aksonie) ma nieliniowy charakter. Bardzo często opisuje się go funkcją sigmoidalną:

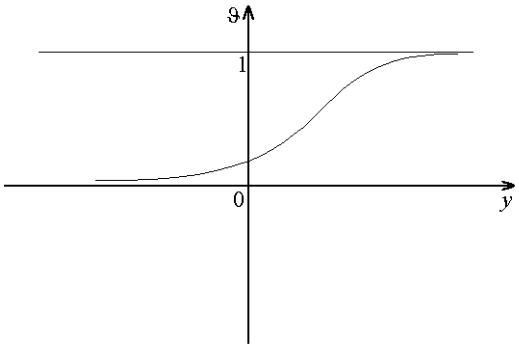
$$\vartheta(x_i) = \frac{1}{1 + e^{-\beta \sum_{i=1}^{n} x_i w_i + w_0}},$$

gdzie w_i są tzw. wagami synaptycznymi, a β współczynnikiem aktywacji, określającym, na ile łatwo jest pobudzić neuron. Jeśli przyjąć

$$y = \sum_{i=1}^{n} x_i w_i + w_0 ,$$

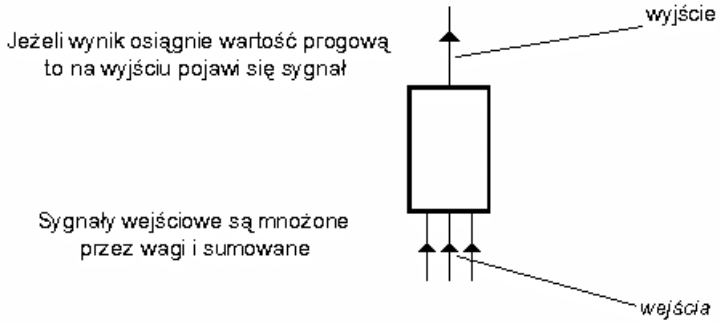
to wykres $\vartheta(y)$ wyglądać będzie tak, jak to przedstawia Rys. 2.

Zależność ta jest wygładzoną wersją funkcji Heaviside'a i opisuje swoisty „biologiczny przełącz-



Rys. 2. Funkcja aktywacji neuronu

nik", działający z pewną zwłoką i przez to raczej daleki od elektrycznego ideału. W największym skrócie można zatem powiedzieć, że neuron funkcjonuje w sposób zobrazowany na Rys. 3.

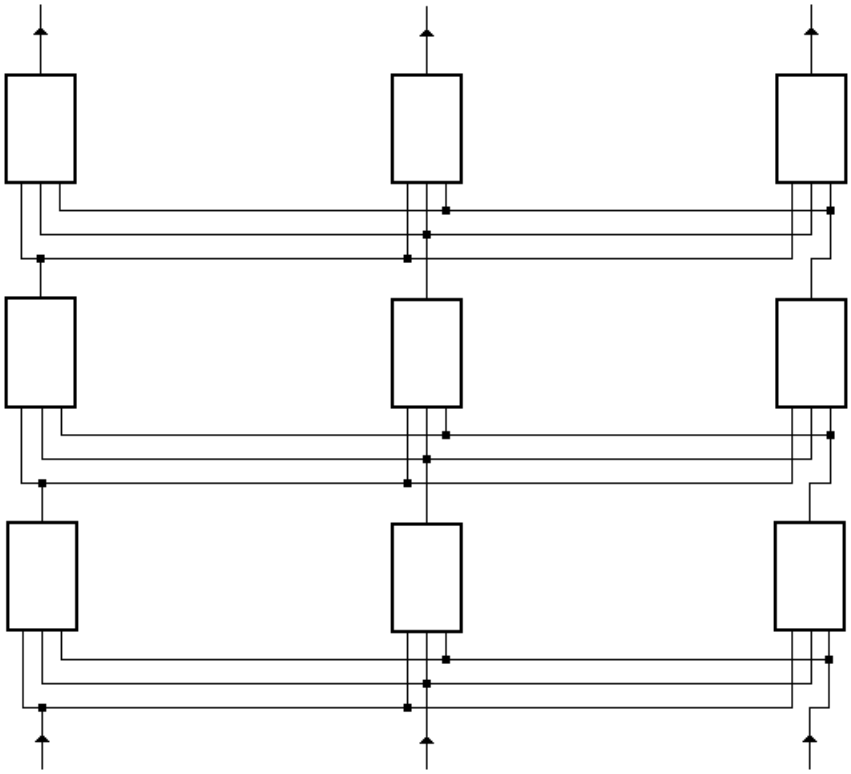


Rys. 3. Pojedynczy neuron

Sygnały wejściowe są „ważone", czyli skalowane liczbami wi. Dendryty, których wagi są większe, mają też większe znaczenie, czyli w większym stopniu decydują, czy neuron zostanie pobudzony, czy też nie. Istota rzeczy polega jednak na tym, że wagi mogą się zmieniać. Stroiąc je odpowiednio, jesteśmy w stanie nauczyć neuron reagować mocniej na impulsy pochodzące z wybranych dendrytów. Fakt ten staje się niezwykle użyteczny, jeśli pojedyncze neurony połączymy

w większą strukturę – w sieć. Z jednej strony powiększymy wtedy pojemność informacyjną układu (więcej wag), z drugiej zaś – już przy trzech warstwach – będziemy w stanie zrealizować praktycznie każdą zależność funkcyjną między wejściem a wyjściem (por. [Tade93]).

Jedną z możliwych topologii sieci, zwaną perceptronem, przedstawia Rys. 4.



Rys. 4. Perceptron trójwarstwowy

Dobierając odpowiednio wagi synaptyczne poszczególnych neuronów możemy diametralnie zmieniać działanie całego układu. Właśnie od wartości wag zależy, jakie konfiguracje sygnałów wejściowych będą sieć pobudzać, a jakie zostaną przez nią zignorowane. Specyficzny, łagodny przebieg funkcji aktywacji neuronu (Rys. 2) sprawia, że sieć wykazuje się pewną tolerancją, tzn. jeśli reaguje na określoną konfigurację sygnałów, to – nieco mocniej lub słabiej – reaguje też na konfiguracje w jakiś sposób do niej podobne. Przypomina to (bynajmniej nie przez przypadek!) mechanizm abstrakcyjnego myślenia. Mówiąc „pomarańcza" nie odnosimy się do jednej, konkretnej pomarańczy, ale do bardziej ogólnego bytu, który matematyk nazwałby klasą abstrakcji znanych nam pomarańcz ze względu na określoną relację równoważności, opisującą podobieństwo owoców. Abstrahujemy więc od tego, czy pomarańcze są małe, czy duże, mniej lub bardziej słodkie, soczyste itd., a skupiamy się na istocie bycia pomarańczą. Podobnie działać będzie sieć z Rys. 4: odpowiednio trenowana skoncentruje się na najistotniejszych cechach danej konfiguracji sygnałów, pomijając cechy drugorzędne. Zwróćmy uwagę na określenie „odpowiednio trenowana". Oznacza ono, że to od nauczyciela (trenera) zależy, czego nauczy się sieć. Nic dziwnego, w końcu jeżeli małemu dziecku będziemy powtarzać, że ów soczysty owoc o pomarańczowej skórce nazywa się „orange", nie zaś „pomarańcza", przyjmie ono to do wiadomości, zapamiętując po prostu nazwę w obcym języku. Mechanizmy abstrahowania nie ulegną jednak zmianie, jako że są one wynikiem samej architektury mózgu, lub – jak w naszym przypadku – sztucznej sieci neuronowej.

3. Uczymy sieć

Proces trenowania sieci neuronowej przebiega w niezwykle naturalny sposób. Zaczynamy od losowo dobranych wartości wag synaptycznych, następnie podajemy na wejścia sieci określony sygnał i sprawdzamy, jak bardzo sygnał wyjściowy różni się od tego co chcemy uzyskać. Z początku będzie on miał oczywiście jakąś przypadkową wartość. Różnicę odpowiedzi sieci i naszych oczekiwań (ściślej – moduł różnicy) nazwiemy błędem odpowiedzi sieci. Zależy on oczywiście od konfiguracji wag synaptycznych całego układu, w związku z czym można postrzegać go jako funkcję

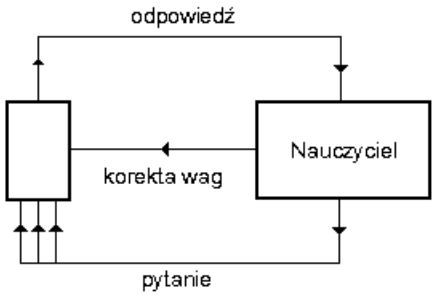
$$\mathsf{E} = \mathsf{E}(w_1, w_2, \ldots w_n),$$

określoną na *n*–wymiarowej przestrzeni tych wag, czyli, ujmując rzecz geometrycznie – jako (*n-1*)–wymiarową hiperpowierzchnię. Wytrenowanie sieci neuronowej sprowadza się więc ostatecznie do znalezienia minimum takiej funkcji, czyli zdaje się być dobrze określonym matematycznie, a w dobie komputerów – również rozwiązywalnym technicznie zagadnieniem. W tym miejscu chcielibyśmy sobie pozwolić na dwie uwagi.

Po pierwsze, w praktycznych zastosowaniach ilość wag synaptycznych jest całkiem duża, może wynosić np. kilkadziesiąt tysięcy, a to z jednej strony oznacza określone trudności obliczeniowe, z drugiej zaś całkowicie przekreśla możliwość wyobrażenia sobie kształtu funkcji błędu. Właśnie tak należy rozumieć popularne stwierdzenie, że sieci neuronowe działają jak czarne skrzynki, tzn. nie wiadomo, co dokładnie robią. Co robią – oczywiście wiadomo, według jakich zasad – też. W każdej chwili możemy poznać wartości wszystkich wag, wyznaczyć funkcję błędu itd., rzecz jednak w tym, że nic z tego nie zrozumiemy, podobnie jak neurochirurg badający ludzki mózg niewiele dowie się na temat świadomości pacjenta, mimo że pozornie dysponuje wszystkimi możliwymi danymi.

Po drugie, algorytmy poszukiwania ekstremów funkcji wielu zmiennych zazwyczaj dość łatwo znajdują minima lokalne. Z takiego lokalnego „dołka" często trudno się wydostać i trafić do innego, który może być głębszy. W literaturze często spotyka się fałszywe naszym zdaniem stwierdzenie, że dobrze wytrenować sieć neuronową, znaczy znaleźć globalne minimum funkcji błędu. Wprowadza się w związku z tym specjalne metody numeryczne, które, choć nie gwarantują sukcesu, ewidentnie zwiększają jednak prawdopodobieństwo znalezienia minimum globalnego. O ile można się zgodzić z tezą, że pierwsze z brzegu minimum lokalne niekoniecznie jest szczególnie dobre, o tyle trudno znaleźć rozsądne uzasadnienie stwierdzenia, że na pewno dobre będzie minimum globalne! Zapomnijmy bowiem na chwilę o zawiłościach numeryki i wróćmy do podstaw, tzn. do przyczyny dla której szukamy ekstremum. Chodzi nam przecież nie o to, żeby sieć idealnie dopasowała się do wzorca, ale aby posiadła umiejętności uogólniania, czyli abstrakcyjnego „myślenia". Można przypuszczać – a wieloletnie doświadczenia autora uzasadniają taki pogląd – że zbyt głębokie minimum pozbawia sieć takich możliwości. Przeuczona sieć przypomina człowieka, który za pomarańcze uzna tylko owoce pochodzące np. z Izraela, natomiast te z Kalifornii odrzuci z uwagi na drobne różnice w wyglądzie, czy smaku. Podobnie zresztą niedouczona sieć jest jak dziecko, które mandarynkę nazwie „taką małą pomarańczą".

Co zatem zrobić, aby nasza sieć naprawdę zaczęła abstrakcyjnie „myśleć"? Przyjrzyjmy się jednej z praktycznych metod uczenia układów neuropodobnych, tzw. metodzie wstecznej propagacji błędów (backpropagation). Nie wdając się w matematyczne szczegóły, które można znaleźć np. w [HKP93, Tade93, Osow96], chcielibyśmy jedynie stwierdzić, że polega ona na rzutowaniu błędu sieci wstecz, na poszczególne warstwy i w oparciu o otrzymane wyniki – korygowaniu wag pojedynczych neuronów, tak jak to przedstawia Rys. 5. Całą operację powtarza się oczywiście cyklicznie aż do uzyskania zadowalających rezultatów.



Rys. 5. Uczenie neuronu

Ten koncepcyjnie prosty i dość skuteczny algorytm znajduje oczywiście minima lokalne, ale wsparty pewnymi dodatkowymi pomysłami – nie kończy na pierwszym z brzegu. W świetle rozważań z poprzedniego akapitu można przypuszczać, że ważny jest np. kształt dołka z (lokalnym) minimum. Intuicja podpowiada na przykład, że powinien on być zarówno głęboki, jak i stosunkowo szeroki. Czy jednak można mieć dobrą intuicję w przestrzeni liczącej kilkadziesiąt tysięcy wymiarów? To pytanie pozostawimy bez odpowiedzi. Ograniczymy się jedynie do stwierdzenia, że scharakteryzowanie punktu pracy sieci, jako takiego, w którym zeruje się pierwsza różniczka funkcji błędu, a druga jest dodatnio określona, uważamy za stanowczo niewystarczające. Należałoby zatem pomyśleć o jakiejś lepszej geometrycznej charakterystyce tego miejsca. Jest to zapewne interesujący kierunek badań, wymagający wielu pomysłów i jeszcze większej ilości pracy, niestety bez gwarancji sukcesu.

Inne podejście do zagadnienia bazuje na pewnej analogii z termodynamiką fenomenologiczną. Układ taki jak gaz doskonały z jednej strony posiada szalenie skomplikowaną mikroskopową dynamikę, a pomysł, aby badać ruch każdego z atomów wydaje się absurdalny. Z drugiej strony, gaz można opisać za pomocą tzw. parametrów zewnętrznych, takich jak temperatura, ciśnienie, czy entropia. Parametry te nie tylko w zupełności nam wystarczają, ale stanowią wręcz dokładnie tę charakterystykę, której potrzebujemy, podczas gdy znajomość położeń i pędów wszystkich atomów i tak nie na wiele by się zdała z uwagi na niestabilność rozwiązań. Podobnie można opisywać sieć za pomocą pewnych parametrów zewnętrznych. Jednym z nich (ale nie jedynym!) będzie bez wątpienia błąd popełniany w każdym cyklu nauki, innym może być błąd wyznaczany na zestawie testowym, tzn. takim, którego nigdy nie użyto podczas uczenia sieci. Posługując się naszym przykładem z pomarańczami, to tak jakby minimalizować błąd popełniany przez sieć przy rozpoznawaniu owoców z Izraela, a potem sprawdzać jak poradzi ona sobie z tymi z Kalifornii. Umiejętna minimalizacja obu błędów powinna przynieść sensowne rezultaty.

Przedstawiona powyżej idea nie jest wcale nowa, jednak przeglądając literaturę można odnieść wrażenie, że nie odkryło jej dla siebie jeszcze wielu autorów. Podobnie zresztą zdają się nie być jej świadomi twórcy sporej części dostępnych na rynku narzędzi neuronowych. Implementacja wcale nie należy zresztą do zadań trywialnych, a przede wszystkim wymaga zreformowania pewnego sposobu myślenia o oprogramowaniu neuropodobnym. Jednym z systemów praktycznie wykorzystujących koncepcję testowania sieci równolegle z ich trenowaniem, rozbudowaną do tzw. metody RDO (Reduced Data Optimization) autorstwa Biura Matematyki Stosowanej, jest stworzony przez tę właśnie firmę *N-Predictor*, o którym nieco więcej w rozdziale 5.

4. Sieć jako model

Wbrew niektórym obiegowym opiniom działanie sieci neuronowej w żadnym stopniu nie sprowadza się do „bezmyślnego” wpasowywania krzywej w punkty pomiarowe, co skrytykowaliśmy w rozdziale 1. Przeciwnie, jego istotą jest modelowanie. Nie narzucamy sieci z góry żadnej zależności funkcyjnej, a jedynie przekazujemy jej materiał „do przemyśleń”. To sieć tworzy na-

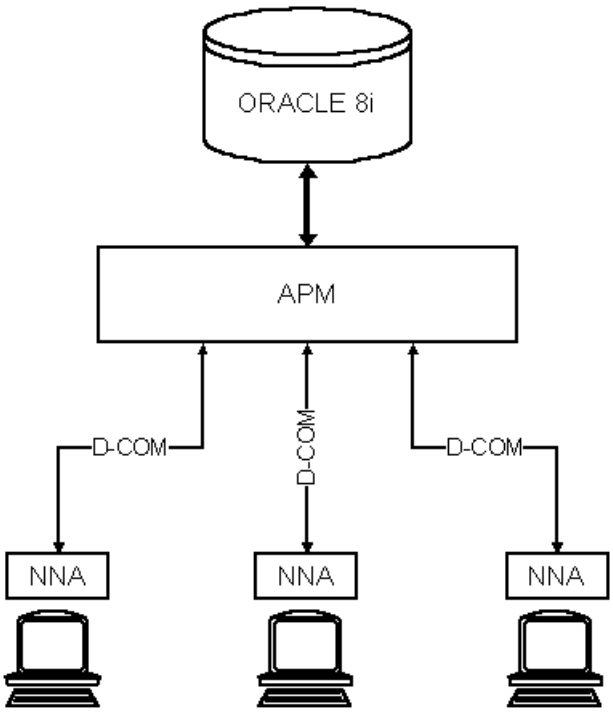
stępnie model rzeczywistości, tyle tylko, że model ten jest całkowicie niezrozumiały dla człowieka, tzn. nie da się go opisać słowami, ani też przedstawić w postaci formuł matematycznych. Stanowi to oczywiście pewien dyskomfort, jednak, jeśli sieć działa poprawnie, to znaczy, że model jaki sobą przedstawia jest dobry. Jak już wspominaliśmy, „czarna skrzynka" może nam zresztą odpowiedzieć, co robi i w jaki sposób, tyle, że w niezrozumiałym dla nas języku. Co nam po kilkudziesięciu tysiącach wag – liczb zmiennoprzecinkowych, w których zapisany jest model? Zgodnie z wcześniejszymi rozważaniami, to mniej więcej tak, jakbyśmy chcieli posiąść wiedzę naukowca badając budowę jego mózgu, albo podziwiać muzykę Mozarta dysponując wydrukiem z rejestratora natężeń i częstotliwości fal dźwiękowych emitowanych przez poszczególne instrumenty!

Problem polega więc w istocie na trudności w komunikacji pomiędzy dwoma rodzajami inteligencji, naturalną i sztuczną. Sztuczna robi swoje, ale nie potrafi naturalnej wytłumaczyć (w języku tej ostatniej) jak to robi. Objaśnienie sposobu postępowania ewidentnie wymaga wspięcia się na wyższy poziom abstrakcji niż samo postępowanie. Nie wiemy jakim bilardzistą okazałby się sir Isaac Newton, gdyby wszedł do któregoś ze współczesnych pubów, jednak przypuszczamy, że niewielu bywalców owych lokali byłoby w stanie samodzielnie sformułować zasady dynamiki.

5. *Syndis Energia* i *N-Predictor*

Aby nie pozostać całkowicie w sferze teorii, pragniemy teraz omówić pewien przykład zastosowania sieci neuronowych do prognozowania. Zgodnie z obietnicą zawartą w tytule dotyczyć on będzie energetyki, a konkretnie prognozowania zapotrzebowania na energię elektryczną. Dlaczego prognozowanie takie jest potrzebne? Dlatego, że nie umiemy niestety skutecznie magazynować prądu, co oznacza, że w zamkniętym systemie energia wytworzona musi się bilansować z zużytą (włączając w to straty przy przesyle, transformowaniu napięć itp.). Z drugiej strony żadna elektrownia nie jest w stanie na bieżąco dostrajać mocy produkcyjnej do obciążenia. Z przyczyn technologicznych nie da się błyskawicznie zmieniać ilości wytwarzanej pary, a zatem zawsze trzeba utrzymywać pewną nadwyżkę, którą w zależności od potrzeb albo zasili się turbinę albo „puści w gwizdek". Oczywiście, im mniejsza nadwyżka, tym większe oszczędności. Zgodnie z obowiązującym prawem energetycznym (więksi) odbiorcy i dystrybutorzy zawierają kontrakty na dostawy określonej ilości energii po określonej cenie. Jeśli zużyją jej więcej niż zamówili, za nadwyżkę muszą zapłacić znacznie drożej. Podobnie, jeśli nie zużyją zamówionej energii, i tak muszą za nią zapłacić. W ich interesie leży zatem stworzenie możliwie dokładnych prognoz zapotrzebowania na energię. A kwoty potencjalnych oszczędności są tu niemałe, bo np. dla średniej wielkości zakładu energetycznego w skali miesiąca mogą one sięgnąć kilkudziesięciu milionów złotych.

Jednym z dostępnych na rynku rozwiązań wspomagających gospodarowanie energią elektryczną jest stworzony przez firmę Mikronika z Poznania system *Syndis-Energia*. Przy budowaniu prognoz wykorzystuje on oprogramowanie *N-Predictor*, autorstwa krakowskiego Biura Matematyki Stosowanej (integrację przeprowadzono przy wsparciu funduszu PHARE 2000 PL 0003.07.05-2 Krajowy program rozwoju MSP „Innowacje i technologie dla rozwoju przedsiębiorstw"). Poszczególne moduły *N-Predictora* znalazły też zastosowanie w systemie PORSENNA i nieco bardziej szczegółowo zostały opisane w [Wyro00]. Przypomnijmy więc, że ważną cechą całego rozwiązania jest wykorzystanie koncepcji obliczeń rozproszonych (grid computing), które w praktyce realizuje się na szeregu zwykłych komputerów klasy PC, wyposażonych w możliwie szybkie procesory. Na poszczególnych maszynach uruchamiane są procesy obliczeniowe *NNA*, z których każdy symuluje jedną (całkiem sporą) sieć neuronową. Efektem treningu jest więc wiele nauczonych sieci – wielu ekspertów, którzy następnie muszą „przedyskutować" swoje opinie („dyskusja" przybiera tu dość prostą formę uśrednienia wyników, obliczenia estymatora błędu i innych podobnych zabiegów matematycznych). Całość koordynuje specjalny nadzorca *APM*, który rozdziela zadania, zbiera wyniki i komunikuje się z bazą danych (Rys. 6). Połączenia z ekspertami realizowane są w technologii D-COM, przy czym każdy z partnerów musi być tu zarówno klientem jak i serwerem usługi.



Rys. 6. Architektura rozwiązania *N-Predictor*

Aby zwiększyć efektywność działania całego układu, sieci poddawane są ustawicznej kontroli, tzn. przez cały czas bada się ich zdolności prognostyczne testując poprawność odpowiedzi na zestawach danych, których nigdy nie były uczone. Trening składa się z dwóch etapów. Pierwszy można porównać do nauki w szkole podstawowej, gdzie wymaga się niewiele, tzn. pozwala na popełnianie chwilami dość sporych błędów. Potem przychodzi egzamin i tylko te sieci, które zaliczą go prawidłowo, mogą uczyć się dalej, podczas gdy pozostałe się likwiduje, a zwolnione komputery – wykorzystuje do dalszych obliczeń. Drugi etap nauki odbywa się już pod stałą kontrolą, a parametry uczenia automatycznie zmieniane są na bieżąco, odpowiednio do postępów sieci. Wygląda to trochę tak, jakby nadzorca ustalał dla każdego z potencjalnych ekspertów indywidualny tok studiów. Jeśli jednak student sobie nie radzi, jest bezwzględnie usuwany z uczelni, aby zrobić miejsce następnemu, lepiej rokującemu koledze.

Jakich danych używa się do uczenia sieci? Punktem wyjścia dla tworzenia dowolnych prognoz musi być zawsze identyfikacja czynników x_1, $x_{2, \ldots}x_n$, od których zależy prognozowana wartość. W przypadku zużycia energii przez odbiorców indywidualnych, będą to przede wszystkim temperatura powietrza, nasłonecznienie, informacja o dniu tygodnia (np. zupełnie inaczej wygląda zużycie w niedzielę, a inaczej w poniedziałek), godzinie doby i oczywiście sam czas t, który, nota bene, jako parametr musi pojawić się w przypadku prognozowania dowolnych zjawisk. Mamy zatem wielowymiarową zależność

$$E = E(x_1, x_2, \ldots x_n, t),$$

zgodnie z którą każdy z zestawów danych użytych do uczenia sieci składałby się z czasu, wartości energii w tym czasie oraz wartości pozostałych zmiennych. Dla i-tej chwili (np. doby, godziny) mielibyśmy więc

$$(t_i, E_i, x_{1i}, x_{2i}, \ldots x_{ni}).$$

Można argumentować, że w rzeczywistości każdy z parametrów x (np. temperatura powietrza) również zależy od czasu, co w praktyce uzasadniałoby znacznie prostszą formułę:

$$E = E(t)$$

oraz zestawy postaci

$$(t_i, E_i).$$

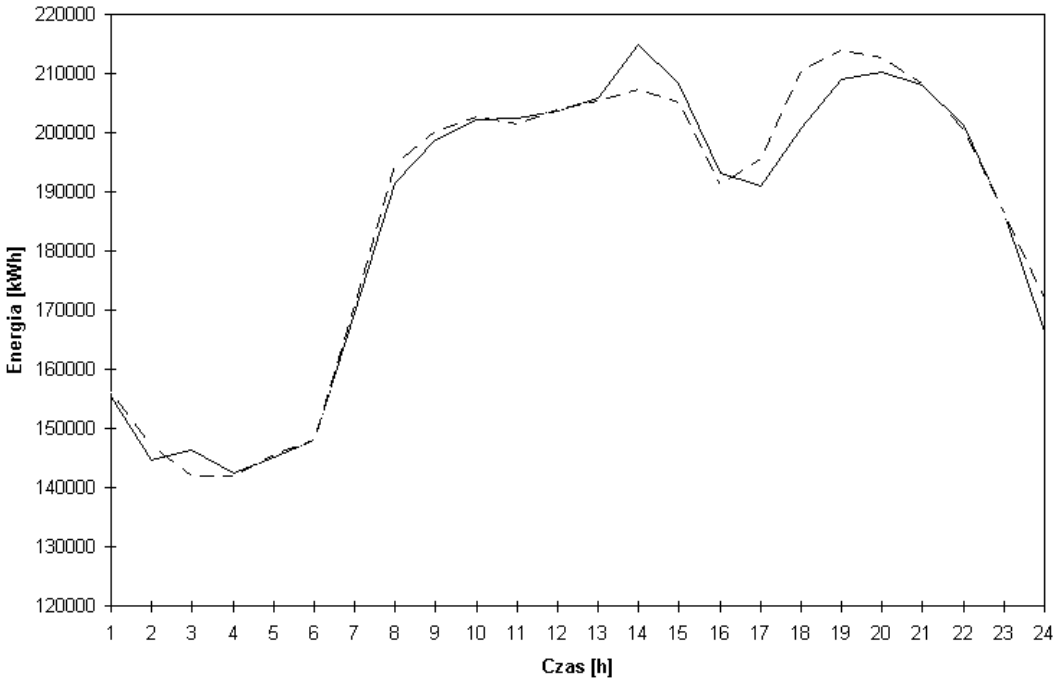
Oznacza to jednak, że nasza sieć w pewien sposób pośrednio musiałaby prognozować również parametry *x*, co miałoby głębszy sens tylko wtedy, gdybyśmy nie mogli przewidzieć ich inaczej. Tylko czy warto prognozować temperaturę powietrza metodami neuronowymi? Raczej nie, bowiem meteorologia dysponuje już całkiem niezłymi modelami, a przewidywaniem pogody w dniu jutrzejszym zajmują się sztaby ludzi wyposażonych w komputery, radary, sztuczne satelity i inne mniej lub bardziej nowoczesne przyrządy. Lepiej więc skorzystać z ich pracy, pozornie komplikując sam problem prognozowania zużycia energii. Jeżeli informacje meteorologiczne włączymy do zestawów uczących, sieć otrzyma więcej danych do nauki, a my zasadniczo będziemy mogli odpowiedzieć na pytanie o jutrzejsze zapotrzebowanie na prąd przy założeniu różnych warunków pogodowych. Wybierzemy rzecz jasna te, które podpowiedzą nam synoptycy.

Oczywiście trzeba mieć świadomość, że im więcej danych, tym większa musi być sieć, aby mogła pomieścić niezbędne informacje, zaś im większa sieć, tym dłuższy czas potrzebny jest do jej wytrenowania. W praktyce może to więc narzucać spore ograniczenia.

Jak zauważyliśmy już wcześniej, sieci neuronowej pod żadnym pozorem nie wolno przeuczać, gdyż wtedy traci ona swoją umiejętność abstrahowania. Oznacza to, że sieć wytrenowana do prognozowania zużycia energii w pewnym okresie nie może zostać „douczona" dla innego okresu. Można albo jej użyć, wychodząc z założenia, iż warunki zewnętrzne nie uległy istotnym zmianom, albo skorzystać z innej, trenowanej od nowa. Jeśli szkoda nam trochę dobrego eksperta, możemy jednak go sklonować, tzn. wykorzystać w nowym procesie nauki początkowe wagi synaptyczne, które w innej sytuacji okazały się już dobre. Bazujemy zatem na zdolnościach wrodzonych, tworząc wierną replikę nowo narodzonego geniusza i szkoląc go ponownie, jednak tym razem do wykonania innego zadania. Rozwinięciem takiego podejścia jest wykorzystane w oprogramowaniu *N-Predictor* mieszanie materiału genetycznego, tzn. tworzenie krzyżówek zdolnych sieci w celu dalszego ich uczenia. Cóż, jak widać, informatycy bezustannie czerpią z ideału, którym niezmiennie okazuje się Natura...

6. Efekty

Chcielibyśmy na koniec przedstawić pewien rzeczywisty rezultat działania sieci neuronowej. Wykres z Rys. 7 obrazuje porównanie prognozy zapotrzebowania na energię dla poszczególnych godzin wybranej doby z realnym zużyciem na terenie obsługiwanym przez jeden z polskich zakładów energetycznych. Zależność rzeczywista została przedstawiona przy pomocy linii ciągłej, a prognozowana – przerywanej, przy czym średnie odstępstwo od rzeczywistości wynosi 1,31%, zaś maksymalne – 4,87%. Wynik wydaje się dość dobry, tym bardziej, że uzyskano go przy bardzo uproszczonym podejściu, gdzie (z powodu braku danych) pominięto istotne informacje meteorologiczne. W celu zwiększenia pojemności informacyjnej sieć rozbudowano sztucznie do rozmiarów dwukrotnie większych niż było to wymagane z uwagi na strukturę danych. Jak się okazało, wpłynęło to pozytywnie na jakość prognoz, lecz dalsze powiększanie układu nie dawało już specjalnych efektów, a tylko wydłużało czas uczenia. Sam proces nauki przebiegał równolegle na 10 komputerach i trwał ok. 16 godzin. Każdą sieć uczono w 50 tysiącach cykli, zaś ostatecznie w głosowaniu wzięło udział 48 ekspertów. Oznacza to, że nauczenie jednego eksperta kosztowało średnio ponad trzy godziny pracy jednego komputera, choć wydaje się, że ilość cykli nauki i wspomniany koszt można zredukować. Zmniejszanie liczby ekspertów prowadzi, jak łatwo zgadnąć, do spadku jakości prognoz, a zależność ma charakter w przybliżeniu eksponencjalny. I tak na przykład dla 24 ekspertów średni błąd wynosi 1,36%, dla 12 – 1,41%, a dla 6 – już 1,50%.



Rys. 7. Porównanie prognozy z zużyciem rzeczywistym

Niestety, nie zawsze otrzymuje się tak dobre wyniki. Czasem prognozy bywały gorsze, a czasem – przy zadanej dokładności – nie uzyskiwało się ich wcale. Takie sytuacje, jako szczególnie niepokojące, wymagały oczywiście dokładnego przeanalizowania. Jak się okazało, miały one miejsce, gdy na krótko przed okresem, którego dotyczyła prognoza, dochodziło do jakichś poważnych zaburzeń, np. awarii w systemie energetycznym, bądź u któregoś ze znaczących odbiorców przemysłowych, co znajdując odzwierciedlenie w zestawach uczących, wywoływało dezorientację trenowanej sieci i w efekcie uniemożliwiało stworzenie dobrej prognozy. Eksperci uparcie powtarzali „nie wiem" i należało przyjąć to do wiadomości. Można było oczywiście skłonić ich do udzielenia jakiejś odpowiedzi zmniejszając (pośrednio) wymaganą dokładność przewidywań, lecz to, siłą rzeczy, prowadziłoby tylko do ekspertyz pozbawionych sensu. Fakt ów skłania nas zresztą do wypowiedzenia na koniec ważnego stwierdzenia, o którym, mimo całej jego oczywistości, zdaje się wciąż zapominać wiele osób:

Tego co nieprzewidywalne z definicji nie da się przewidzieć

Dedykujemy je zarówno tym, którzy korzystając z różnych, również neuronowych narzędzi, tworzą prognozy bez zastanowienia się nad istotą zjawiska, jak i tym, którzy bezkrytycznie w te prognozy wierzą.

Bibliografia

[HKP93] Hertz, J., Krogh, A., Palmer, R.G.: Wstęp do teorii obliczeń neuronowych, Wydawnictwa Naukowo-Techniczne 1993, ISBN 83-204-1680-9.

[Osow96] Osowski, S.: Sieci neuronowe w ujęciu algorytmicznym, Wydawnictwa Naukowo-Techniczne 1996, ISBN 83-204-1976-X.

[Tade93] Tadeusiewicz R.: Sieci neuronowe, Akademicka Oficyna Wydawnicza RM 1993, ISBN 83-85769-03-X.

[Wyro00] Wyrozumski T.: Prognozowanie neuronowe w oparciu o dane ekonomiczne z baz Oracle, VI Konferencja PLOUG, Zakopane 2000, ss. 297-304.