

# Prognozowanie neuronowe w oparciu o dane ekonomiczne z baz Oracle

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

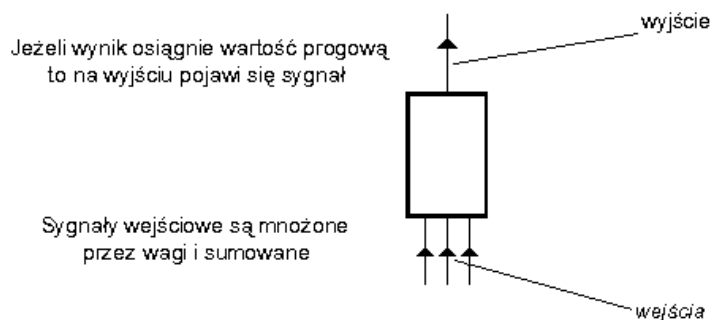
**Abstrakt.** Prezentujemy zastosowanie sztucznych sieci neuronowych do prognozowania pewnych wskaźników ekonomicznych, związanych z rynkiem kapitałowym. Omawiany system wykorzystuje dane gromadzone w bazach Oracle, a sieci neuronowe trenuje się w schemacie wielostopniowym, z użyciem metody RDO. W celu uzyskania lepszych wyników stosuje się algorytmy genetyczne. Istotny wzrost mocy obliczeniowej zapewnia przetwarzanie równoległe z wykorzystaniem stacji roboczych pracujących w całej sieci lokalnej użytkownika, przy czym komunikacja międzyprocesowa opiera się na technologii D-COM. Do poprawnej parametryzacji systemu niezbędne jest kontrolowanie funkcjonowania sieci neuronowych, co z kolei oznacza konieczność zwrotnego rejestrowania i analizowania dużej ilości danych. Do tego celu wykorzystuje się również bazy Oracle.

## 1. Specyfika sieci neuronowych

Celem tego referatu jest zaprezentowanie pewnego realnego zastosowania sieci neuronowych w większym systemie, znanym pod handlową nazwą *PORSENNA DSS (Portfolio Selection based on Neural Network Advisor)* i służącym do konstruowania i zarządzania portfelami papierów wartościowych. Nie będziemy się jednak koncentrować ani na opisie tego oprogramowania ani na problematyce ekonomicznej stanowiącej punkt wyjścia, lecz raczej na praktycznych aspektach użycia sieci neuronowych, rzeczywistych trudnościach, na jakie się przy tym napotyka i skutecznych sposobach ich przezwyciężania. Mamy nadzieję, że przedstawione tu obserwacje okażą się pomocne wszystkim, którzy zechcą w tworzonych przez siebie aplikacjach zastosować układy neuropodobne, korzystając przy tym np. z pakietu Oracle Darwin. Ze względu na obszerność problematyki skoncentrujemy się tylko na jednym rodzaju sieci, tzw. perceptronach nieliniowych (szczególnie interesujących w zagadnieniach prognozowania), choć szereg uwag będzie miało ogólniejszy charakter. Być może uda nam się przy tym rozwiązać pewne mity i pomóc słuchaczom przekonać ich przyszłych klientów do nowej, ważnej technologii informatycznej.

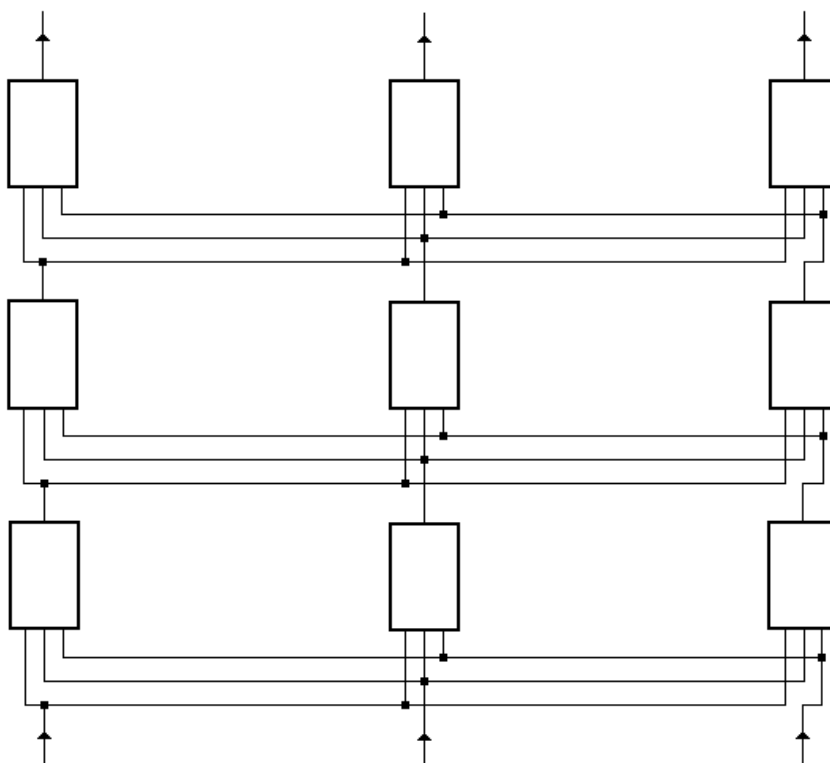
Czym są sztuczne sieci neuronowe? W największym skrócie - są układami przetwarzającymi informację w sposób równoległy, wzorowanymi na ludzkim mózgu. Choć analogia jest (przynajmniej na dzień dzisiejszy) dość daleka, to jednak sieci wykazują zadziwiająco wiele cech, o które posądza się raczej myślące organizmy żywe niż tradycyjne krzemowe komputery. Można więc mówić o ich zdolnościach, intuicji, rozkapryśzeniu, czy nawet zwykłym lenistwie.

Istotą sieci neuronowych jest, jak wiadomo, możliwość ich uczenia, polegająca w rzeczywistości na długotrwałym dostrajaniu dużej ilości liczb ważących przetwarzane sygnały, zwanych wagami synaptycznymi (rys. 1).



Rys. 1. Pojedynczy neuron

Nauczona sieć to układ o wielu wejściach i wielu wyjściach (rys. 2), który na określone sygnały wejściowe odpowiada we właściwy sposób i może w związku z tym stanowić model pewnego zjawiska - fizycznego, biologicznego, ekonomicznego itp.



Rys. 2. Perceptron trójwarstwowy

Oczywiście, najciekawsze jest to, jak sieć reaguje w tych sytuacjach, których nie została dokładnie wyuczona, a więc jak potrafi uogólniać, ekstrapolować, czy wręcz prognozować. Ponieważ odpowiedzi sieci neuronowych mają zawsze charakter przybliżony (i to raczej niekontrolowanie przybliżony), nie ma najmniejszego sensu używać ich tam, gdzie dysponujemy ścisłym modelem matematycznym (np. przy opisie elementarnych zjawisk fizycznych). Niekiedy jednak zdarza się, że z jednej strony ze względu na stopień skomplikowania zjawiska modelowanie matematyczne praktycznie nie wchodzi w rachubę, a z drugiej istnieją poważne przesłanki, by sądzić, że zjawisko nie przebiega w sposób chaotyczny. Sytuacja taka występuje np. w ekonomii, gdzie bardzo często podejrzewa się istnienie związków pomiędzy pewnymi wskaźnikami, jednak charakter tych związków nie jest do końca jasny, a w dodatku może się zmieniać w czasie. W takich sytuacjach sieci neuronowe mogą okazać się bardzo pomocne.

Oczywiście, szczególną pokusę rodzi prognozowanie rozwoju sytuacji na rynkach papierów wartościowych, a konkretnie – przepowiadanie kursów akcji na następny dzień. Istnieje szereg znanych i nieznanymi aplikacji, które mają służyć temu celowi oraz – zapewne - jeszcze więcej osób, które próbują się nimi posługiwać. Nie wdając się w długie polemiki autor, który jako fizyk z wykształcenia przez wiele lat zajmował się strukturą czasoprzestrzeni, chciałby w tym miejscu wyrazić swoje prywatne głębokie przekonanie, iż przewidzenie przyszłości jest niemożliwe i to niezależnie od tego, czy miałaby to czynić wróżka, czy sieć neuronowa. Teoretycznie możliwe byłoby natomiast stworzenie układu, który odpowiadałby na pytanie, jak zachowa się rynek, jeżeli nie stanie się nic wykraczającego poza ramy określonego modelu, a więc np. jeżeli nie zajdą żadne nadzwyczajne zjawiska, takie jak niespodziewany kryzys lub nieodpowiedzialna wypowiedź wpływowego polityka. Niestety, abstrahując od problemów czysto technicznych (moc obliczeniowa komputerów) – sieci neuronowe niezbyt dobrze nadają się do prognozowania procesów szybko

zmiennych i mało stabilnych. Co zatem potrafią? Otóż znacznie lepiej przewidują zjawiska długookresowe i n.b., są w tym podobne do ludzi. Każdy odpowiedzialny analityk chętniej powie nam, jak ogólnie wyglądać będzie sytuacja na rynku w przyszłym tygodniu, lub miesiącu, niż ile dokładnie zapłacimy jutro za akcje Elektrimu. Powstaje więc pytanie, czy sieci neuronowe w ogóle mogą się tu do czegoś przydać. Odpowiedź brzmi: tak, jeżeli interesuje nas nie tyle spekulacyjna gra na zwwyżki i niżki akcji, co raczej długoterminowe inwestycje. Nie chcąc wdawać się w tym miejscu w szczegółowy opis zastosowań tego rodzaju prognoz, zainteresowanych odsyłamy do bogatej literatury poświęconej analizie portfelowej, natomiast w dalszej części referatu zajmijmy się przede wszystkim informatyczną stroną zagadnienia.

## 2. Sieci - o czym się nie mówi...

Na rynku dostępnych jest bardzo wiele programów, i to zarówno tych „dużych”, jak i pomniejszych, również shareware i freeware, służących do symulowania sieci neuronowych. Ich autorzy prześcigają się w mnożeniu opcji pozwalających na zadawanie dowolnej architektury sieci, kształtów funkcji aktywacji neuronu, algorytmów uczenia itd. Pozornie jest więc w czym wybierać. Pozornie, bo oferowana mnogość możliwości nie oznacza realnych korzyści. Dlatego warto uzmysłowić sobie kilka elementarnych faktów.

I tak po pierwsze, „napisanie” sieci neuronowej nie stanowi specjalnego problemu i nie jest żadnym wielkim wyzwaniem. Należy patrzeć na nie raczej w kategoriach programu na zaliczenie dla dobrego studenta, niż poważnego projektu informatycznego.

Po drugie, uczenie sieci jest procesem bardzo czasochłonnym, a wszelkie warianty oferowane przez program-kombajn oznaczają kolejne instrukcje „if”, wykonywane miliony razy przez procesor. Zatem im więcej opcji w programie, tym większy narzut na czas nauki.

Po trzecie, w przypadku sieci perceptronowych nie ma obecnie żadnej sensownej teorii, która pozwoliłaby, w zależności od problemu skłonić się ku takiej a nie innej architekturze układu lub postaci funkcji aktywacji. W efekcie od autora programu otrzymujemy dużą ilość swobodnych parametrów, które musimy dostroić prowadząc długie i żmudne testy, a to akurat jest najtrudniejsze i – w kategoriach uczelnianych – mogłoby być raczej tematem na dobrą pracę magisterską niż na zaliczenie.

Szczęśliwy nabywca taniego pakietu neuronowego musi mieć świadomość, że otrzymał bardzo niewiele, a większość pracy dopiero przed nim i dlatego wybierając oprogramowanie „z półki” należy zwracać uwagę nie na mnogość rodzajów sieci, jakie oprogramowanie to oferuje, lecz raczej na udostępnione narzędzia kontrolne i analityczne, przykłady, dokumentację itp.

W tym miejscu dochodzimy do podstawowej kwestii: jak w ogóle zmierzyć jakość sieci neuronowej? W bardzo prostym, działającym modelu prognozowania zachowań rynku papierów wartościowych, badanym przez firmę BMS, mamy już do czynienia z dość dużą ilością wag synaptycznych, przekraczającą 20 tysięcy. Proces uczenia sieci sprowadza się więc z matematycznego punktu widzenia do poszukiwania ekstremów funkcji w przestrzeni wektorowej liczącej ponad 20 tysięcy wymiarów. Oczywiście zawodzą tu intuicje, a wszelkie próby zrozumienia, „co naprawdę robi sieć”, wydają się bezsensowne. Układ przypomina trochę gaz składający się z dużej ilości atomów. Wiadomo, że z praktycznego punktu widzenia nie jest istotne, jakie są położenia i prędkości poszczególnych atomów, lecz raczej, jak kształtują się wielkości termodynamiczne, a więc temperatura, ciśnienie, entropia itp. Podobnie, złożony system przetwarzający informację można próbować opisać za pomocą kilku parametrów, zdających sprawę z jego stanu. Niestety, wielu autorów upraszcza sprawę dostrzegając tylko jeden taki parametr, tzn. błąd uczenia – różnicę pomiędzy rzeczywistą a oczekiwaną odpowiedzią na zadany zestaw wejściowy. Ten błąd poddawany jest, jak wiadomo, minimalizacji podczas procesu trenowania sieci. Czy jednak można powiedzieć, że dobra sieć, to sieć charakteryzująca się bardzo małym błędem ucznia? Raczej nie, gdyż naprawdę nie chodzi o to, by sieć dobrze reagowała na to, czego jej nauczono, lecz raczej, by udzielała sensownych odpowiedzi na te pytania, których jej nigdy nie

postawiono podczas treningu! Dopiero wtedy okazuje się ona twórcza i może między innymi sensownie prognozować.

Stwierdzenie, iż sieć typu perceptronowego powinna być opisywana nie jednym, lecz kilkoma globalnymi parametrami jest kluczowe z punktu widzenia stworzonego przez nas oprogramowania. W toku prowadzonych prac udało nam się wprowadzić trzy użyteczne parametry: błąd uczenia, niedoskonałość i zgodność. Nie wdając się w szczegóły konstrukcyjne stwierdzimy tylko, że poprawianiu jakości sieci odpowiada spadek dwóch pierwszych parametrów do wartości bliskich zera, podczas gdy trzeci parametr powinien zbliżać się do jedynki. Ta obserwacja pozwala nie tylko oceniać sieci, ale też skutecznie sterować ich trenowaniem, w znaczący sposób poprawiając jego efektywność. Cała metoda, stworzona przez firmę BMS, określana jest mianem *RDO (Reduced Data Optimization)*.

### 3. Model dwustopniowej edukacji

Sieci neuronowe, podobnie jak ludzie, mogą posiadać pewne wrodzone uzdolnienia lub predyspozycje. Chodzi o to, że proces uczenia rozpoczyna się od wylosowania początkowych wartości wag i tu można mieć mniej lub więcej szczęścia. Na złe wagi początkowe są zasadniczo dwa sposoby: albo opracować takie wyrafinowane metody uczenia (sterowanie parametrami), które nawet z najmniej zdolnego materiału uczynią geniusza, albo też pozbyć się tych wag i losować dalej w nadziei, że znajdzie się lepsze. O ile pierwsza metoda wydaje się ciekawsza, to jednak myśląc o badaniu wag i dobieraniu parametrów musimy mieć cały czas na uwadze fakt, iż poruszamy się w przestrzeni mającej wiele tysięcy wymiarów, więc „zobaczenie” czegokolwiek jest bardzo trudne. Druga metoda została całkiem skutecznie przetestowana na ludziach. Powszechnie uważa się bowiem, że studia wyższe nie są dla wszystkich, lecz tylko dla najzdolniejszych, najpracowitszych i najambitniejszych (niekiedy i najbogatszych, lecz jeśli inne z wymienionych cech nie idą w parze z tą ostatnią, to i tak skutki są mizerne). Być może więc również i sztuczne sieci neuronowe powinny być w pewien sposób kwalifikowane do nauki – najlepiej w modelu edukacji stopniowej, czyli najpierw kształcenie podstawowe, a dopiero potem studia (etap szkoły średniej został z pewnych powodów wyeliminowany). I tak w podstawówce sieć nie jest ściśle kontrolowana – ma dużą swobodę, może się trochę uczyć, trochę bawić, trochę leniuchować, a odpowiednie wykresy przypominają zachowanie rozbrykanego dziecka. To wszystko jest tolerowane, jednak na koniec przychodzi egzamin. Jeżeli sieć go nie zda, jest likwidowana, jeżeli zda – uczy się dalej, ale pod czujnym okiem profesora. Sprawdza on cały czas, czy student nie znalazł się aby w szkole wyższej przez przypadek i czy nauka przebiega prawidłowo. Tego procesu absolutnie nie należy mylić z tzw. uczeniem nadzorowanym, znanym z literatury o sieciach neuronowych i wykorzystywanym również i tutaj. Posługując się znów analogią z klasycznym systemem oświaty można powiedzieć, że uczenie nadzorowane polega tylko na odpytywaniu ucznia na lekcji, podczas gdy metoda opisana powyżej – dodatkowo na wprowadzeniu okresowych prac kontrolnych, czy egzaminów, ocenianych przez niezależną komisję według obiektywnych i surowych kryteriów, sprawdzających nie tylko bieżące wiadomości studenta, lecz również jego samodzielność i kreatywność. Później, mając kilka dobrych sieci, możemy dla ostatecznego rozwiązania problemu powołać „komisję ekspertów”, a więc uśrednić otrzymane wyniki oraz przeprowadzić pełną statystyczną analizę błędów, która pozwoli oszacować dokładność prognozy.

### 4. Dlaczego Oracle?

Odpowiedź na to pytanie jest krótka i prosta: Oracle, a ściślej, Oracle 8i - ponieważ potrzebowaliśmy dobrej bazy danych. Jak widać, przedstawiona w największym zarysie metoda trenowania sieci neuronowych jest dość złożona i bardzo odbiega od tego, co można spotkać w programach klasy shareware. Jej skuteczność zależy od zbierania i sprawnego przetwarzania dużych ilości informacji związanej nie tylko z tym, czego się uczy, ale – i to przede wszystkim – jak się uczy. Konieczne jest permanentne rejestrowanie różnych charakterystyk procesu nauki i analizowanie ich, zarówno on-line, automatycznie, jak i post factum, przez człowieka, który od

czasu do czasu powinien zbadać przebiegi celem ewentualnego poprawienia parametrów. Trzeba mieć więc szybki dostęp do danych i oczywiście gwarancję niezawodności, gdyż utrata wyników może kosztować miesiące pracy. Z powodów, które szczegółowo wyjaśniamy w następnym paragrafie, program komunikujący się z bazą musi ponadto uruchamiać szereg wątków, z których każdy niezależnie sięga do danych. To również stanowi pewne wyzwanie dla bazy i bez wątpienia Oracle radzi sobie z nim doskonale.

Odrębną kwestią jest bezpieczeństwo informacji. W sytuacji, gdy wyniki pracy sieci mogą w najmniejszy choćby sposób przekładać się na decyzje dotyczące miliardowych inwestycji, nie można sobie pozwolić na to, aby dane nie były odpowiednio zabezpieczone przed ingerencją ze strony niepowołanych osób. W istocie, w omawianym systemie *PORSENN*A wyróżnia się kilka ról ze zróżnicowanymi prawami dostępu do poszczególnych tabel. Oracle oferuje odpowiedni system zabezpieczeń i co więcej, skrupulatnie rejestruje wszystkie operacje dokonywane przez użytkowników. Dzięki temu można więc w razie potrzeby można ustalić kto i kiedy zmodyfikował np. pewne istotne dla systemu parametry, a przez to przyczynił się do dodatkowych zysków lub spowodował straty.

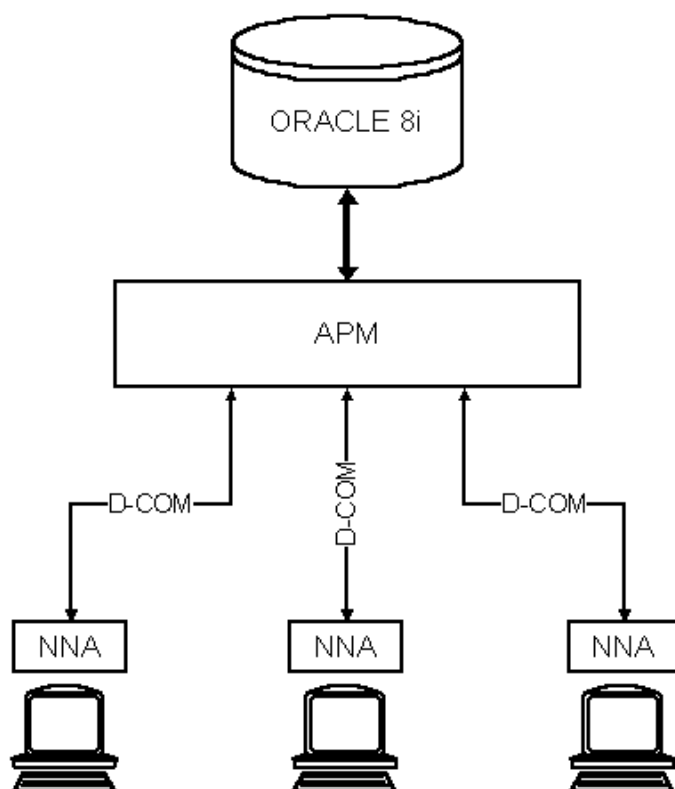
Jak wiadomo, cała wiedza sieci neuronowej kryje się w wagach synaptycznych, a więc skuteczne jej przechowanie wymaga zarejestrowania bardzo długiego (kilkadziesiąt tysięcy) ciągu liczb zmiennoprzecinkowych. Oczywiście, zapisywanie ich w kolejnych rekordach, jedna po drugiej, jest czasochłonne i nie ma najmniejszego sensu, jako że motor bazy wykonuje przy każdym pojedynczym zapisie cały szereg skomplikowanych operacji. Dużo lepszym rozwiązaniem jest natomiast przekształcenie ciągu liczb w łańcuch znaków i zapamiętanie go jako pojedynczego obiektu. Warto przy tym przez chwilę zastanowić się nad ilością miejsc znaczących każdej z wag, aby bez potrzeby nie zapamiętywać zbyt dużej porcji informacji. Otóż wbrew pozorom ilość ta wcale nie musi być wielka, gdyż sieci neuronowe ze swej natury nie są specjalnie „precyzyjne” i podobnie jak ludzie przedkładają oszacowania nad dokładne obliczenia. N.b., dzięki temu są też dość odporne na uszkodzenia: sieć w której zupełnie zmieniono kilka wag na ogół nie przestaje działać, tak samo jak człowiek, który utracił pewną ilość komórek nerwowych w mózgu (w istocie codziennie tracimy ich całkiem sporo!). W rzeczywistych modelach ekonomicznych badanych przez firmę BMS i stosowanych przez jej klientów pojedynczy zestaw wag synaptycznych liczy sobie w praktyce kilkaset kilobajtów. Do zapisu takich obiektów Oracle oferuje typ CLOB, który zresztą doskonale spełnia swoje zadanie: proces rejestrowania i odczytywania wag przebiega szybko i sprawnie nie powodując zbędnych przestoju w systemie.

## 5. Technologia obliczeń rozproszonych

W jaki sposób w praktyce zrealizować scenariusz dwustopniowej edukacji *RDO*, tak aby w sensownym czasie otrzymać sporą ilość dużych, dobrze nauczonych sieci neuronowych? Ogólnie są dwie metody. Pierwsza, oczywista, polega na wykorzystaniu maszyn obliczeniowych wielkiej mocy, przy czym decydująca jest tu nie pamięć czy rozmiar dysków, lecz całkiem po prostu – szybkość procesorów. Wadą tego podejścia są stosunkowo duże nakłady na sprzęt, bardzo podrażające całe rozwiązanie. Warto też pamiętać, że technologia sieci neuronowych wciąż jeszcze budzi sporą nieufność i na ogół trudno jest przekonać klienta do zakupu superkomputera i to jeszcze zanim pokaże mu się jakieś sensowne wyniki. Dlatego wybrano alternatywne podejście, polegające na zrównolegleniu obliczeń. Jest to bardzo naturalne, jeżeli z założenia trenujemy nie jedną, lecz wiele sieci, odrzucamy te złe, wybieramy najlepsze, a potem uśredniamy ich prognozy. Mając do dyspozycji wiele komputerów klasy PC, pracujących w lokalnej sieci, można rozdystrybuować pomiędzy nie zadania i zdalnie sterować wykonaniem. Z praktycznego punktu widzenia obliczenia można przeprowadzać tylko na komputerach, przy których nikt fizycznie nie pracuje, ponieważ bardzo poważnie obciążają one procesor. Wspomniany system *PORSENN*A potrzebuje wyników nie częściej niż raz w tygodniu, wobec czego idealnym czasem dla ich uzyskania jest weekend. Łatwo policzyć, że niewielka lokalna sieć (np. departamentu inwestycyjnego banku lub funduszu) złożona z pięciu komputerów jest w stanie zaoferować tygodniowo około 300 godzin obliczeń, a to już jest niemało.

Oczywiście cały proces musi być wysoce zautomatyzowany, gdyż jakakolwiek obsługa podczas trenowania sieci raczej nie wchodzi w rachubę. W praktyce system został wykonany w technologii D-COM (choć równie dobrze mogłaby być to CORBA), przy pomocy narzędzi programistycznych firmy Borland. Całość składa się z trzech programów napisanych w Delphi: nadzorcy APM (*Automatic Process Manager*), doradcy NNA (*Neural Network Advisor*) i specjalnego pomocnika (*NNA Listener*). Sam motor sieci neuronowych NNE (*Neural Network Engine*) to niewielka biblioteka DLL, stworzona z uwagi na optymalizację pod kątem szybkości działania, a także możliwość łatwego przeniesienia na inne platformy, w czystym języku C.

D-COM zapewnia obustronną łączność pomiędzy nadzorcą, a studentami NNA (tak podczas procesu uczenia określa się neuronowych doradców), natomiast bezpośredni kontakt z bazą Oracle ma jedynie nadzorca (rys. 4). On pobiera dane, przygotowuje je i przekazuje studentom w postaci tzw. zestawów uczących, odczytuje z bazy informacje sterujące oraz monitoruje przebieg uczenia i zapisuje jego wyniki. Zadaniem nadzorcy jest też oczywiście wyłączanie studentów, którzy uczą się źle, tak aby dać szansę kolejnym kandydatom. Z technicznego punktu widzenia każdemu procesowi uczenia odpowiada odrębny wątek w programie nadzorującym, przy czym poszczególne wątki komunikują się z bazą danych całkowicie niezależnie.

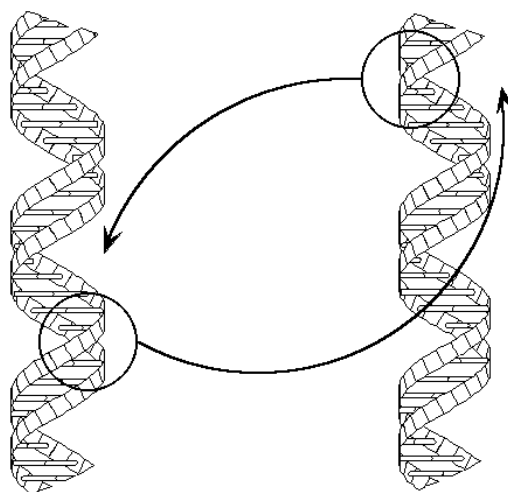


Rys. 3 Architektura obliczeń rozproszonych

Proces trenowania sieci kończy się o określonej godzinie (zazwyczaj w poniedziałek rano) i wtedy jego wyniki są automatycznie przetwarzane. Jeżeli jest więcej nauczonych sieci, a więc więcej ekspertów, to wtedy, tak jak już wspomniano, przeprowadza się wśród nich głosowanie, tzn. średniuje wynik, analizuje błędy i ewentualnie odrzuca sieci, których odpowiedzi przekraczają statystycznie wyznaczony margines tolerancji. Te ostatnie można potraktować jak złych studentów, którzy mimo wszystko zdołali się przecisnąć przez sito systemu edukacji.

## 6. Wrodzone zdolności i genetyka

Ostateczną weryfikacją jakości prognoz powinno być oczywiście porównanie ich z rzeczywistością, co na ogół może nastąpić dopiero po upływie pewnego czasu (kilku dni) od zakończenia nauki. Wśród wielu nauczonych sieci znajdują się takie, które prognozowały lepiej i takie, które były w tym gorsze. Ocenę należy jednak przeprowadzać ostrożnie. Powiedzmy bowiem, że większość naszych sieci przewidywała zmianę pewnego wskaźnika ekonomicznego o 2%, a wskaźnik w istocie zmienił się o 5%. Jeżeli jedna z sieci przewidziała 4,5 %, to może to znaczyć dwie rzeczy: albo była naprawdę dobra, albo udało się jej to przez przypadek. Jeżeli istnieją nie budzące wątpliwości przesłanki, że zmiana wskaźnika o 5% była wynikiem zaburzenia, którego nie dało się przewidzieć (np. trzęsienia ziemi w ważnym ekonomicznie rejonie świata), to oczywiście zachodzi drugi przypadek. Jeśli jednak rozwój wydarzeń był przewidywalny, to raczej mamy do czynienia z pierwszym i wtedy szkoda byłoby takiej sieci nie wykorzystać ponownie. Jak to zrobić? Wydaje się, że wystarczyłoby uczyć ją dalej w przyszłym tygodniu, lecz ten pomysł nie jest najlepszy, gdyż długotrwały trening sieci neuronowych prowadzi do złych rezultatów: sieci przeuczają się i co prawda doskonale odtwarzają zapamiętane wzorce, to jednak przestają być kreatywne i w szczególności tracą zdolność prognozowania. Kluczem do rozwiązania problemu jest odpowiedź na pytanie, dlaczego jedna z wielu identycznie uczonych sieci okazała się wyraźnie lepsza od innych. Otóż po prostu była zdolniejsza „od urodzenia”. Należałoby więc zapamiętać początkowe, wylosowane wagi synaptyczne i sklonować sieć, czyli użyć tych wag ponownie, ucząc ją (już na nieco innych danych) w przyszłym okresie. Takie podejście nie jest wprawdzie złe, lecz nie daje perspektyw na rozwój. Powiedzmy, że nasza sieć była zdolna, ale czy nie znalazłaby się jeszcze zdolniejsza? Klonując sieć w nieskończoność zaprzepaścimy szansę na dokonywanie poprawek, co z pewnością nie jest dobre. Znacznie lepszy mechanizm wynalazła sama natura, a my nazwaliśmy go ewolucją. Wystarczy spojrzeć więc na zestaw wag synaptycznych jak na łańcuch chromosomów - materiał genetyczny, który z pokolenia na pokolenie (z tygodnia na tydzień) może podlegać mutacjom oraz mieszanii w drodze krzyżowania sieci, tak jak pokazano to na rys 4.



Rys. 4 Operacje genetyczne

W rzeczywistości operacje genetyczne wykonywane są przez moduł zarządzający APM raz na pokolenie, czyli raz na jedną sesję obliczeń. Oznacza to modyfikowanie materiału genetycznego np. co tydzień i pozwala realizować długookresowe doskonalenie sieci neuronowych, dokładnie tak, jak ma to miejsce w przypadku prawdziwej ewolucji biologicznej.

## 7. Korzyści

Naszkiecowana powyżej architektura systemu informatycznego znalazła, jak już wspomniano zastosowanie praktyczne w oprogramowaniu służącym do zarządzania portfelami, a także doczekała się pierwszego wdrożenia. Oczywiście sieci neuronowe nie stały się dla klienta maszynką do zarabiania pieniędzy (na to zresztą nie liczył), ale przyniosły mu pewne realne korzyści, podobne do tych, które osobie umiejącej liczyć przynosi kalkulator. Nie jest on postrzegany jako cudotwórca wykonujący magiczne operacje na liczbach, lecz raczej jako użyteczny sługa, biorący na siebie operacje żmudne i czasochłonne. Uważamy, że tak właśnie należy przedstawiać potencjalnym nabywcom aplikacje wykorzystujące sieci neuronowe. Nie powinno się ich zwodzić fałszywymi obietnicami, lecz raczej z góry wyjaśnić, że sieci, konstruowane na podobieństwo mózgu, na pewno nie wymyślą nic więcej niż on. Mogą natomiast okazać się pożyteczne w taki sam sposób, jak wszystkie inne rozwiązania informatyczne, a więc pomagać ludziom w tym, co zasadniczo mogliby zrobić sami i oszczędzać ich bezcenny czas.