Modelarea şi evaluarea impactului investițiilor directe asupra pieții muncii şi evoluțiilor macroeconomice. Metodologie pentru estimarea unor modele neuronale[1]

- cerințe privind seriile de date -

Corina Sâman[2]

Rezumat

Tehnica rețelelor neuronale poate fi aplicată în estimarea unor modele neliniare în problema clasică de regularizare când nu dorim sau nu putem să specificăm forma funcției neliniare. Flexibilitatea acestei tehnici permite oprirea la un model adecvat dintr-o serie de modele analizate. Metoda early-stoping asigură focalizarea pe modele de prognoză.

Introducere

Rețelele neuronale liniare au fost aplicate cu succes în găsirea unor soluții aproximative pentru o largă varietate de probleme, dintre care problema clasică de regularizare (sau estimare ne-parametrică) în care o funcție necunoscută $y = f(\vec{x})$ trebuie aproximată pornind de la o mulțime discretă de date $(\vec{x}_t, y_t)$.

Experiența a arătat că arhitecturi simple, cu relativ puține unități computaționale (noduri) pot avea foarte bune performanțe – totuşi, cu cât numărul de variabile din model este mai mare, alegerea arhitecturii poate avea o importanța foarte mare în obținerea unui model adecvat.

Avantajele folosirii rețelelor neuronale în problema clasică de regularizare sunt: capacitatea de a procesa date de dimensiune mare, abilitatea de a extrage informație semnificativă din datele procesate şi posibilitatea de a lua în considerare faptul că datele mai recente pot avea influență mai mare decât cele mai depărtate în timp.

Rețele neuronale liniare RNL – scurtă prezentare

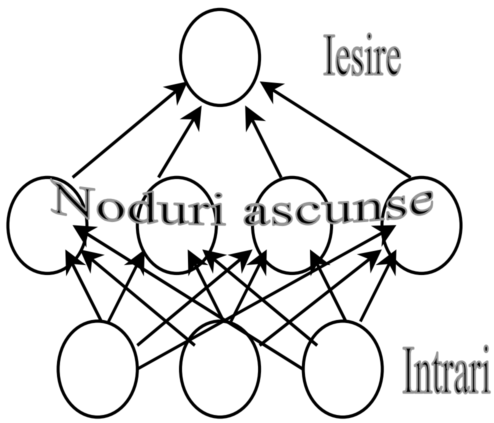
În ultimii 25 de ani s-au dezvoltat tehnici pentru modelarea relațiilor statistice neliniare care sunt denumite rețele de învățare, unele dintre acestea fiind rețelele neuronale liniare

[1] Prezentată în cadrul programului: “Modelarea si evaluarea impactului investitiilor directe nationale si internationale asupra pietii muncii si evolutiilor macroeconomice din Romania”, contract MEC 91-052/ 10 sept 2007, faza decembrie 2007: Etapa I: Documentare stiintifica, crearea bazei informationale, rafinare si analiza exploratorie a datelor.

[2] Institutul de Prognoză Economică, Academia Română, Bucureşti

(multilayer perceptons), cel mai popular tip de rețele, care își au originea în modelele simple ale sistemului nervos biologic.

Formularea generală a unei RNL cu ieșire unică (un singur nod output) și un nivel de noduri ascunse poate fi scrisă:



$$\tilde{f}(\vec{x}) = h(\sum_{i=1}^{k} \alpha_i h(w_{i0} + \vec{w}_{i1}'\vec{x}) + \alpha_0) \quad (1),$$

unde $h(.)$ este o funcție netedă, crescătoare, ca, de pildă, funcția tangentă hiperbolică, $\{\alpha_i\}$ și $\vec{w}$ sunt coeficienți și k este numărul de noduri ascunse.

Pentru o mulțime dată de intrări $\{\vec{x}_t\}$ și valori de ieșire $\{y_t\}$, găsirea unui model RNL presupune estimarea a (d+1)k parametrii $\{w_{i0}\}$ și $\{\vec{w}_{i1}\}$ și k+1 parametrii $\{\alpha_i\}$, în total (d+2)k+1 parametrii.

Specificarea modelului

Pot fi construite RNL (Rețele Neuronale Liniare) ca soluții aproximative pentru problema clasică de regularizare în care o funcție necunoscută $y = f(\vec{x})$ trebuie aproximată pornind de la o mulțime de date discretă $(\vec{x}_t, y_t)$. Vectorul de dimensiune d $\vec{x}_t$ poate fi considerat vectorul variabilelor independente, explicative, y_t variabila dependentă și $f(.)$ funcția neliniară căutată, care să aproximeze valoarea așteptată y_t pentru o intrare $\vec{x}_t$, $y_t \approx f(\vec{x}_t)$.

Problema de regularizare (sau estimare ne-parametrică) poate fi privită ca minimizarea unei funcții obiectiv: $H(f) = \sum_{t=1}^{d} \left\| y_t - \hat{f}(\vec{x}_t) \right\|^2$ (2).

Căutarea unui model neliniar prin tehnica rețelelor neuronale are avantajul flexibilității. Prin ajustarea numărului de noduri și nivele obținem o varietate de funcții neliniare care pot fi luate în considerare în alegerea modelului neliniar optim.

Aproximarea și eroarea de convergență

RNL, ca orice rețea de învățare, s-a dovedit a avea o proprietate de *aproximare universală*. Cybenko (1988) și Hornik (1989) au demonstrat că RNL cu un nivel de noduri ascunse pot reprezenta cu o precizie arbitrar aleasă clase de funcţii continue, liniare sau neliniare, cu intrări şi ieşiri mărginite.

Această proprietate nu spune însă nimic despre cât de uşor de găsit este o aproximare bună, sau cât de eficientă din punct de vedre computaţional este obţinerea soluţiei. In particular se pune întrebarea dacă numărul de date, dimensiunea mulţimii $(\vec{x}_t, y_t)$, de care avem nevoie pentru a estima parametrii reţelei creşte exponenţial cu mărimea ei – aşa numita cursă a dimensionalităţii. Sunt rezultate care arată că dacă se restrânge complexitatea funcţiei modelate această *cursă a dimensionalităţii* poate fi evitată. Barron (1991) găseşte limite ale ratei de convergenţă a erorii de aproximare în RNL în cazul unor ipoteze asupra netezimii funcţiei de aproximat.

Estimarea parametrilor

Structura unei RNL este descrisă prin numărul de neuroni şi nivele din RNL. Ele determină organizarea funcţiei neliniare modelate. Crescând numărul funcţiilor de transfer (adăugând neuroni şi nivele) se creşte complexitatea şi flexibilitatea RNL.

Tocmai flexibilitatea este o proprietate importantă în aproximarea unei multitudini de relaţii funcţionale dintre intrări şi ieşiri. Şi RNL suficient de complexe sunt capabile să aproximeze funcţii arbitrare suficient de bine.

Algoritmii folosiţi în estimarea RNL sunt cunoscuţi ca "algoritmi de antrenare". Aceşti algoritmi sunt asemănători cu rutinele standard de minimizare folosite, de exemplu, în metoda celor mai mici pătrate aplicată funcţiilor neliniare. Algoritmii de antrenare ajustează iterativ parametrii în direcţia gradientului negativ al erorii.

Totuşi practicile de estimare din RNL diferă de tehnicile econometrice în puncte importante.

Pentru a evita "*overfitting*" algoritmul trebuie oprit inainte de a atinge minimul local. Intuitiv, "overfitting" apare când RNL oferă o aproape-perfectă potrivire pentru datele folosite, dar o slabă predicţie în afara datelor de intrare. RNL sunt susceptibile de acest fenomen tocmai datorită flexibilităţii în aproximarea formelor funcţionale diferite.

Una dintre cele mai folosite tipuri de proceduri de oprire a algoritmului este o procedură de validare pe o submulţime a datelor de intrare(cross-validation). Datele sunt partiţionate într-o mulţime de antrenare şi una de validare. Apoi, algoritmul de antrenare este executat pe mulţimea de antrenare până când MSE (Mean Square Error) începe să crească pe mulţimea de validare – moment care apare înainte de atingerea minimului MSE pe mulţimea de antrenare.

Arhitectura modelelor considerate

Ideea folosirii reţelelor neuronale pentru o mai bună cunoaştere a unor mecanisme economice complexe a fost propusă în literatura ştiinţifică (Salzano 1999, Zhang 2003). Urmărim să obţinem modele neuronale care să descrie relaţiile dintre un set de variabile de intrare şi o variabilă de ieşire care să modeleze investiţiile directe totale.

Datele luate în considerare de model sunt lunare, pentru perioada 2000-2006.

In tehnica reţelelor neuronale cu cât avem mai multe date cu atât mai bine. Invăţarea structurii intrinseci a relaţiei existente între variabila de ieşire şi variabilele explicative necesită un volum cât mai mare de date. Orice învăţare se obţine în timp, intervalul de timp considerat fiind pentru model egal cu dimensiunea datelor de intrare.

Modelul I evaluează impactul investiţiilor directe asupra pieţii muncii.
Vom avea : investiţiile directe totale ca variabilă dependentă y;
variabilele explicative : numărul de şomeri şi/sau salariul mediu.

Modelul II evaluează impactul investiţiilor directe asupra evoluţiei macroeconomice.
Vom avea : investiţiile directe totale ca variabilă dependentă y;
variabilele explicative vor fi indicatorii macroeconomici: PIB şi/sau rata inflaţiei, (eventual soldul balanţei de plăţi, deficitul de cont curent).

Modelul III evaluează impactul pieţii muncii asupra investiţiilor directe.
Vom avea : numărul de şomeri şi/sau salariul mediu, ca variabilă dependentă y;
variabilă explicativă : investiţiile directe totale

Modelul IV evaluează impactul investiţiilor directe asupra evoluţiei macroeconomice.
Vom avea : PIB şi/sau rata inflaţiei, ca variabilă dependentă y;
variabilă explicativă: investiţiile directe totale.

Pentru acurateţea modelării şi pentru ca modelele să devină unele predictive, trebuie avută în vedere influenţa valorilor din trecut ale variabilei modelate asupra valorilor ei actuale. De aceea sistemele vor căuta sa ghicească valorile curente ale variabilei de ieşire prin ajustarea unei funcţii dependente de valorile trecute ale intrărilor şi ieşirilor.

La momentul curent t, ieşirea este determinată de intrările **x** la momente trecute diferite ($t-i_1, \ldots, t-i_m$) si de ieşirea **y** la momente trecute ($t-o_1, \ldots, t-o_n$).

Relaţia obţinută de model este de tipul:

$$y(t+1) = f(x(t+1-i_1),\ldots,x(t+1-i_m), y(t+1-o_1),\ldots,y(t+1-o_m)).$$

Preprocesarea datelor

Datele vor fi **normalizate,** deoarece la valori medii pentru variabilele de intrare şi ieşire foarte diferite, algoritmii vor converge foarte încet apărând probleme care, din punct de vedere matematic, se numesc ill-conditioning.

Daca nu am normaliza datele, convergența spre optim poate să apară după un număr de paşi foarte mare, care depinde de diferența ca ordin de mărime dintre media variabilelor implicate în model, dar şi de valorile inițiale pentru ponderi.

In final se va proceda la un proces invers normalizării pentru a găsi valorile reale ale variabilei de ieşire simulată de rețeaua neuronală.

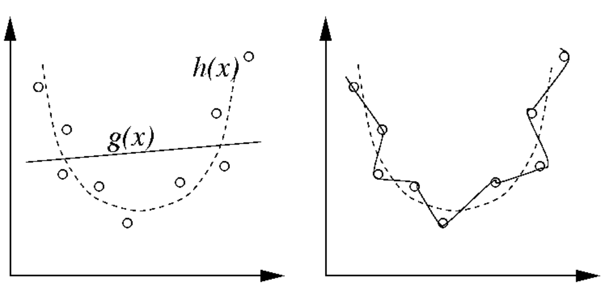
Procedura de antrenare a rețelei neuronale

Proprietatea de *aproximare universală* este cea care dă măsura de flexibilitate a rețelelor neuronale în procesul de învățare a unei structuri şi relații necunoscute. Riscul potențial este apariția problemei de supra-învățare (overfitting), adică antrenarea rețelei pentru a învăța toate particularitățile aflate în eşantionul de date, nu numai a relației reale existente în model.

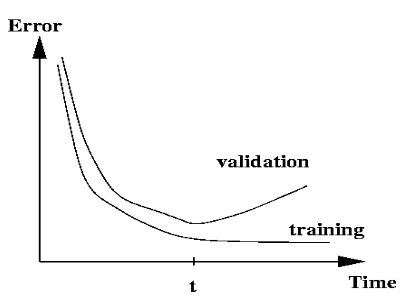
Să ne imaginăm ca avem nişte date inițiale cu o anumită eroare ε :

$$y(x) = h(x) + \varepsilon$$

Vrem să găsim un model g(x) care aproximează h(x).



Modelul liniar g(x) este prea simplu şi are o mare deplasare (bias) față de date, pe când modelul din graficul 2 se potriveşte foarte bine pe intrări (overfitting), dar nu predicționează bine pe h(x).



Obținerea unui echilibru între capabilitatea de învățare şi puterea de generalizare a modelului este foarte importantă în modelarea neuronală, iar soluția este metoda early-stopping.

Se divide spaţiul de intrare în două mulţimi: ***Tr***, o mulţime de antrenare şi ***Val*** o mulţime de validare. Antrenăm reţeaua numai pe mulţimea ***Tr***, dar oprim algoritmul şi testăm reţeaua pe mulţimea ***Val***. Cum ***Val*** este independentă de ***Tr***, performanţa obţinută de model pe mulţimea ***Val*** este o bună măsură de generalizare a modelului (de predicţie).
Atâta timp cât reţeaua învaţă structura datelor (h(x)), performanţa modelului pe mulţimea ***Val*** se va îmbunătăţi . O data ce reţeaua încetează să înveţe structura corecta pentru orice date de intrare şi învaţă numai lucruri adevărate pentru această mulţime de intrare ***Tr***, performanţa pe datele de validare ***Val*** se va deteriora, reţeaua se va opri din trendul de învăţare şi de fapt continuarea procesului de învăţare va duce la potrivirea pe datele eşantionului avut. Pentru a evita această problemă, oprim algoritmul la timpul t, unde performanţa pe mulţimea ***Val*** este optimă. Pentru a evita ca modelul să dea o mare diferenţă între eroarea pe mulţimea de antrenare E_{Tr} faţă de eroarea obţinută pe mulţimea de validare E_{Val} , vom impune şi o condiţie prin care această diferenţă să fie de 10% până la 25% , deci o condiţie de tipul: $\mathbf{E_{Val} < (1 + p)\, E_{Tr}}$.

Este o procedură standard să se antreneze reţelele folosind un număr mare de valori iniţiale aleatoare pentru parametrii şi să se aleagă cea mai bună RNL. Totuşi nu este indicat să se satureze spaţiul parametrilor cu prea multe valori iniţiale, deoarece astfel s-ar găsi minimul global, ceea ce nu este de dorit în tehnica de oprire prematură (early-stopping).

Se vor folosi un număr rezonabil de valori iniţiale aleatore pentru parametrii. Reţeaua se va antrena pentru fiecare set de valori ale parametrilor, urmând să se aleagă cea mai bună soluţie găsită ca model.

In esenţă, tehnica găsirii reţelei iniţiale constă în următorii paşi:

- Se alege o arhitectură a RNL (***k*** noduri ascunse);
- Fiecare arhitectură se antrenează cu metoda *early-stoping* de n ori (n este un număr mic 5<n<10), cu valori aleatoare pentru ponderi şi diferite partiţii ale mulţimii de date în ***Tr*** şi ***Val;***
- Se reduc ponderile cu un factor de scalare γ ($0<\gamma<1$);
- Se reantrenează reţeaua cu noile ponderi;
- Se alege cea mai bună arhitectură RNL faţă de criteriul „cea mai mică eroare dintre valorile simulate şi cele dorite pentru variabila de ieşire”.

Rezultate

Se compară rezultatele obţinute prin cele 4 modele neuronale propuse cu modele de tip VAR corespunzătoare, pentru mai multe orizonturi de previziune. Comparaţia se poate face în ceea ce priveşte R^2 şi MSN (mean square error). Orizonturile de previziune pot fi de 1 lună, 2 luni, un trimestru.
Rezultatele vor permite o discuţie asupra adecvării modelelor obţinute. Avantajele şi dezavantajele utilizării celor două tipuri de modele pot fi analizate sub rezerva

semnificației reduse datorate dimensiunii datelor - relativ mică din perspectiva tehnicii rețelelor neuronale.

Bibliografie

B. Lebaron, A.S. Weigend *A bootstrap evaluation of the effect of data splitting on financial time series*, IEEE Trans. Neural Network, vol.1, nr.9, 1998

F. Girosi , T. Poggio *Networks and the Best Approximation Property*, MIT, paper no.45, 1989

K. Hornik, M. Stinchcombe, H. White *Multilayer feedforward networks are universal approximators*, Neural Networks, vol.2, 1989

V. Kurkovc, M. Sanguineti *Bounds on Rates of Variable-Basis and Neural-Network Approximation*, IEEE Trans. Inform. Theory, vol.47.no.6 Sept.2001

T. Poggio, F. Girosi *Networks for approximation and learning*, Proc. of IEEE vol 78, no.9, 1990

M. Salzano *Neural Networks as tools for increasing the forecast and control of complex systems*, Economics & Complexity, Vol. 2, No. 2, 1999

G. Zang, B.E. Patuwo, M.Y. Hu *Forecasting with artificial neural networks : The state of art*, Int. J.Forecasting vol.14, 1998