

Regulació de temperatura a partir de xarxes neuronals

Les xarxes neuronals són una eina molt útil per a regular la temperatura en diferents aplicacions, des de sistemes de climatització fins a processos industrials.

Una xarxa neuronal pot ser entrenada per a aprendre la relació entre les variables d'entrada (com la temperatura actual, la humitat, etc) i la variable de sortida (la temperatura desitjada). Una vegada entrenada, la xarxa pot predir la temperatura desitjada en funció de les condicions actuals i ajustar els controls del sistema per a aconseguir aquesta temperatura.

Els avantatges de les xarxes neuronals són múltiples ja que són capaces d'aprendre relacions no lineals i complexes, la qual cosa les fa adequades per a sistemes amb comportaments dinàmics i, una vegada entrenades, les xarxes neuronals poden realitzar prediccions ràpidament, la qual cosa permet un control en temps real.

El Departament Tècnic de ACE Automatismes i Control, Elèctric ha realitzat un detallat estudi sobre la Regulació de la temperatura a partir de xarxes neuronals per a la implementació d'un control de temperatura amb un PLC que s'exposa a continuació.

Objectius de l'estudi

Atès que la regulació de la temperatura a la indústria és un element d'ús molt comú però que ens pot complicar en funció dels sistemes, creiem útil la creació d'un objecte programari que pugui auto aprendre.

D'aquesta manera es minimitzaria el temps per a regular els paràmetres del PID i els possibles canvis ambientals que afectin al sistema.

L'objectiu d'aquest estudi és crear una xarxa neuronal artificial per a la implementació d'un control de temperatura amb un PLC. Aquesta xarxa estarà integrada a un bloc de funcions que sigui escalable a altres dispositius.

Dades

Les proves es faran amb un simulador que compte amb una sonda i una resistència calefactors. D'aquesta manera, extraurem les dades de temperatura de la sonda i actuarem sobre la resistència calefactors.

Xarxes neuronals artificials

Les xarxes neuronals consisteixen en una simulació de les propietats observades en els sistemes neuronals biològics a través de models matemàtics recreats mitjançant mecanismes artificials.

El funcionament d'una neurona, consisteix en multiplicar els valors d'entrada (x_1, x_2, x_3, \dots) pels seus respectius pesos (w_1, w_2, w_3, \dots) i sumar-los. Aquest resultat es multiplica per una funció d'activació que dependrà del tipus de resultat que necessitem.

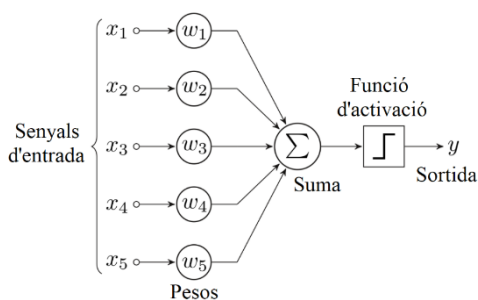


Figura1. Neurona

$$X * W^t = (x_1, x_2, \dots, x_n) * \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \sum_{i=1}^n x_i * w_i$$

Una xarxa neuronal artificial té tres capes en la seva forma més bàsica. La informació flueix d'una capa a una altra com ho faria el nostre cervell. Les diferents capes són:

Capa d'entrada: és el punt d'entrada de les dades del nostre sistema. N'hi haurà tantes com entrades es tingui d'informació.

Capa oculta: són les capes on es processa la informació. Com a més complexa la xarxa, més capes i més neurones hi ha a aquesta capa.

Capa de sortida: és la capa on s'obté el resultat de sortida de tots els càlculs.

Les neurones es van connectant entre elles i van formant una xarxa com la de la figura2 .

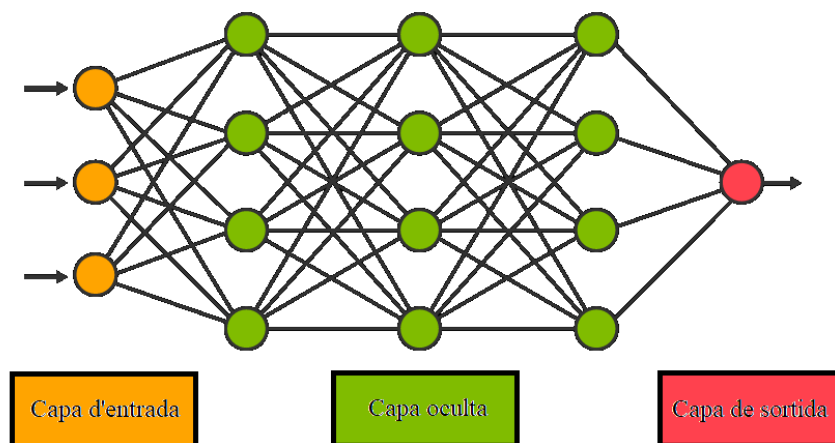


Figura 2. Xarxa neuronal

Modos d'aprenentatge del sistema

El desenvolupament de l'aprenentatge automàtic es centra en tres categories principals: aprenentatge supervisat, aprenentatge no supervisat i aprenentatge profund. Pel nostre treball ens fixarem en el primer i l'últim.

Aprenentatge supervisat

Aquest procés de generació de coneixement es realitza amb un grup d'exemples o dades etiquetades en els que els resultats són coneguts prèviament. Aquest tipus de model aprèn d'aquests resultats i incorpora ajustos en els paràmetres interiors per poder adaptar-se a dades noves que s'introdueixen al sistema.

Mitjançant aquest aprenentatge s'alimenta un conjunt de resultats que permet realitzar prediccions adequades del comportament de les dades noves que encara no han estat processades

Aprenentatge reforçat

Té com a finalitat construir models que augmenten el rendiment aprèn com a base d'un resultat o la recompensa que es genera a partir de cada iteració realitzada. Aquest model utilitza la recompensa com a paràmetre d'ajustament per a accions futures, de tal forma que la nova acció compleixi amb l'acció correcta. D'aquesta manera, per cada procés que realitza es pot veure si ha millorat o no en funció del procés anterior i tenir en compte aquesta dada.

Inici del procés

Es proposa en iniciar amb un aprenentatge reforçat. Aquest pot ser el més adient ja que tenim accés a l'error actual que hi ha entre la temperatura de consigna i la

temperatura actual. D'aquesta manera es poden fer els càlculs de l'error proporcional, integral i derivatiu.

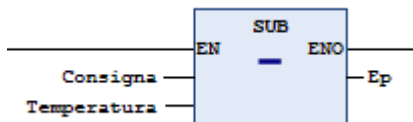
Obtenció de les dades

El treball començarà amb una xarxa neuronal artificial de tres neurones d'entrada (Ep, Ei i Ed), tres neurones ocultes i una de sortida.

Les primeres dades que tindrem del procés serà la temperatura de consigna i la temperatura actual. Treballant amb aquestes dades obtindrem l'error proporcional, l'error integral i l'error derivatiu. Aquestes tres dades seran les entrades de la nostra xarxa neuronal.

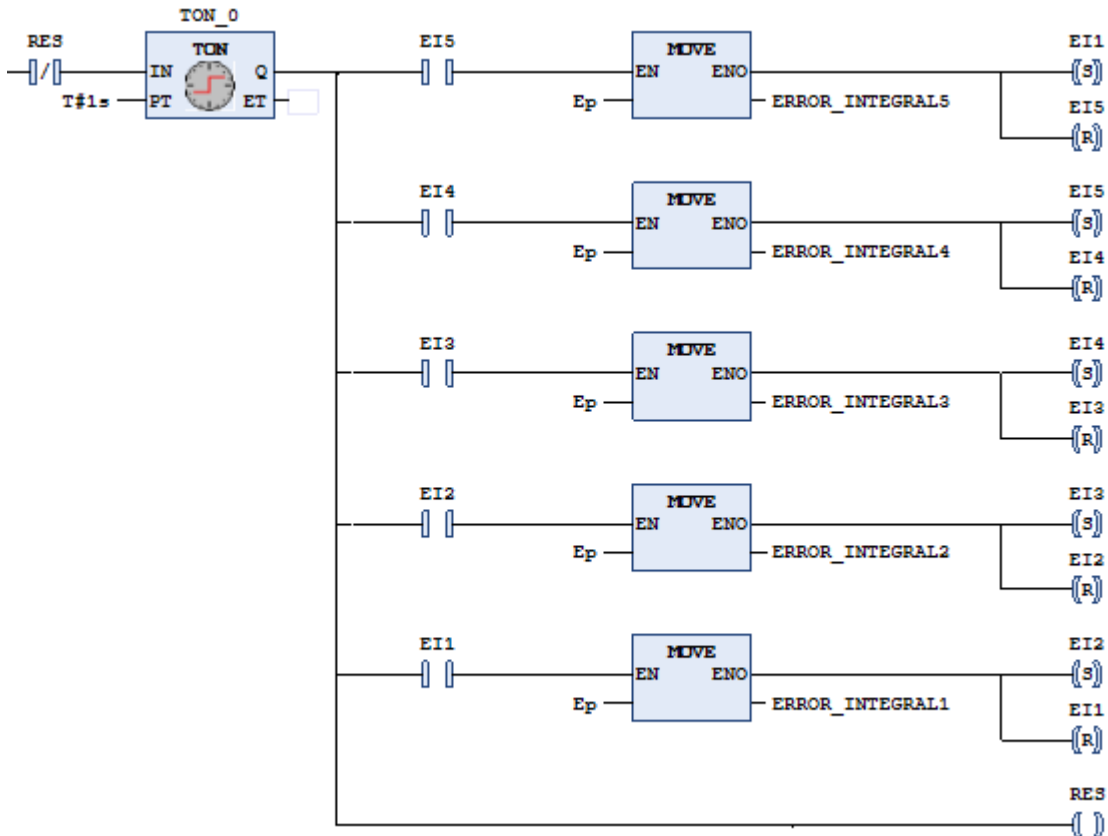
Error proporcional (Ep)

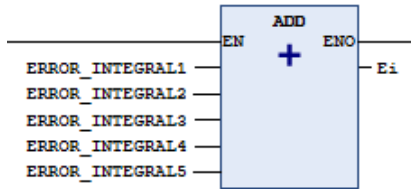
L'obtindrem de la resta entre els valors de la consigna i la temperatura



Error integral (Ei)

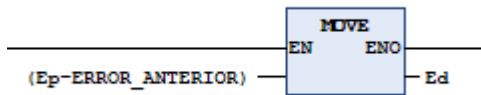
Per obtenir l'error integral, sumarem els valors de l'error proporcional dels últims 5 segons.





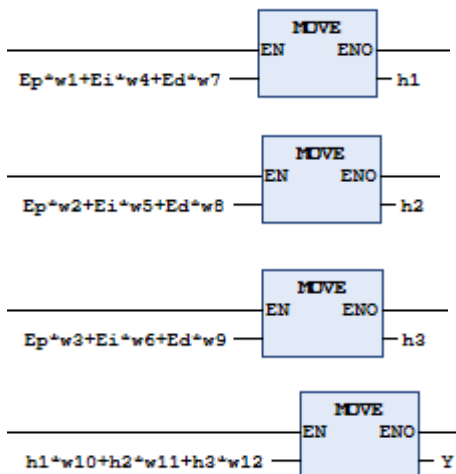
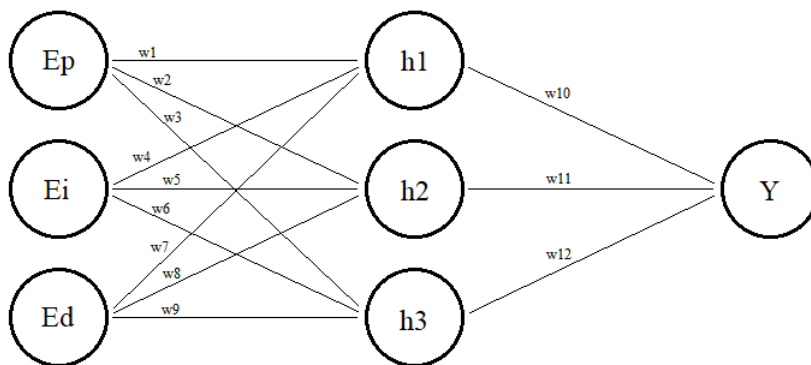
Error derivatiu (Ed)

Calcularem el valor de la pendent que hi ha durant un segon. Per a fer-ho restarem a l'error proporcional actual, el valor de l'error proporcional de fa un segon.



Processament de càlculs neuronals

El càlcul de la xarxa neuronal seguirà el procediment mostrat a la figura1.



Seguint amb els objectius anteriors, es vol crear un programa d'autoaprenentatge per a fer un control de temperatura semblant a un PID. D'aquesta manera reduir el temps per a trobar els paràmetres adequats i fer que variïn en funció de variacions de l'entorn que envolta la màquina.

Es buscarà crear una xarxa neuronal artificial per a la implementació d'un control de temperatura amb un PLC. Aquesta xarxa estarà integrada a un bloc de funcions que sigui escalable a altres dispositius.

En aquest estudi seguirem els següents passos per a la implementació d'un controlador PID utilitzant tècniques d'aprenentatge automàtic

1. **Recopilació de dades:** Primer, serà necessari recopilar dades del sistema actual. Això inclou les entrades al controlador com l'error entre el valor desitjat i el valor actual, les sortides del controlador i qualsevol altre variable rellevant del sistema.
2. **Processament de dades:** Tot seguit cal processar les dades per adequar-les per a l'aprenentatge automàtic. Això pot incloure la normalització de les dades, càlculs inicials, la eliminació del soroll i la identificació de característiques rellevants.
3. **Entrenament del model:** A continuació, cal entrenar un model d'aprenentatge automàtic amb aquestes dades. Per això existeixen diferents models, com les xarxes neuronals, els arbres de decisió o els algorismes de regressió.
4. **Implementació del model:** Una vegada el model estigui entrenat i validat, es pot implementar a un PLC per ajustar els paràmetres del PID a temps real.

1. Recopilació de dades

Aquest primer assaig es farà amb un kit de temperatura que consta d'una resistència calefactorsa i una sonda de temperatura PT100. Les dades i els càlculs s'obtidran a partir d'un PLC.

Les dades que obtindrem en aquesta prova, son la temperatura actual (temperatura_act) i la temperatura a la que es vol arribar (consigna). A partir d'aquestes dades processarem els càlculs corresponents per trobar els errors proporcional, integral i derivatiu per a que siguin l'entrada de la nostra xarxa neuronal.

2. Processament de dades

Es farà els càlculs per a poder obtenir la informació de l'error proporcional, integral i derivatiu del sistema.

Error proporcional

Obtidrem el valor de l'error proporcional a partir de la diferència d'entre la temperatura a la que volem arribar i la temperatura actual

$$u(t) = e(t)$$

$$Ep = \text{consigna} - \text{temperatura}_{act}$$

Error integral

L'error integral és el sumatori de l'error proporcional. Nosaltres farem un sumatori del valor actual de l'error proporcional i dels últims 4. Aquests es prendran en intervals de 1 segon.

$$u(t) = \int_0^t e(\tau) d\tau$$

$$Ei = \sum Ep_{act} + Ep_{act-1} + Ep_{act-2} + Ep_{act-3} + Ep_{act-4}$$

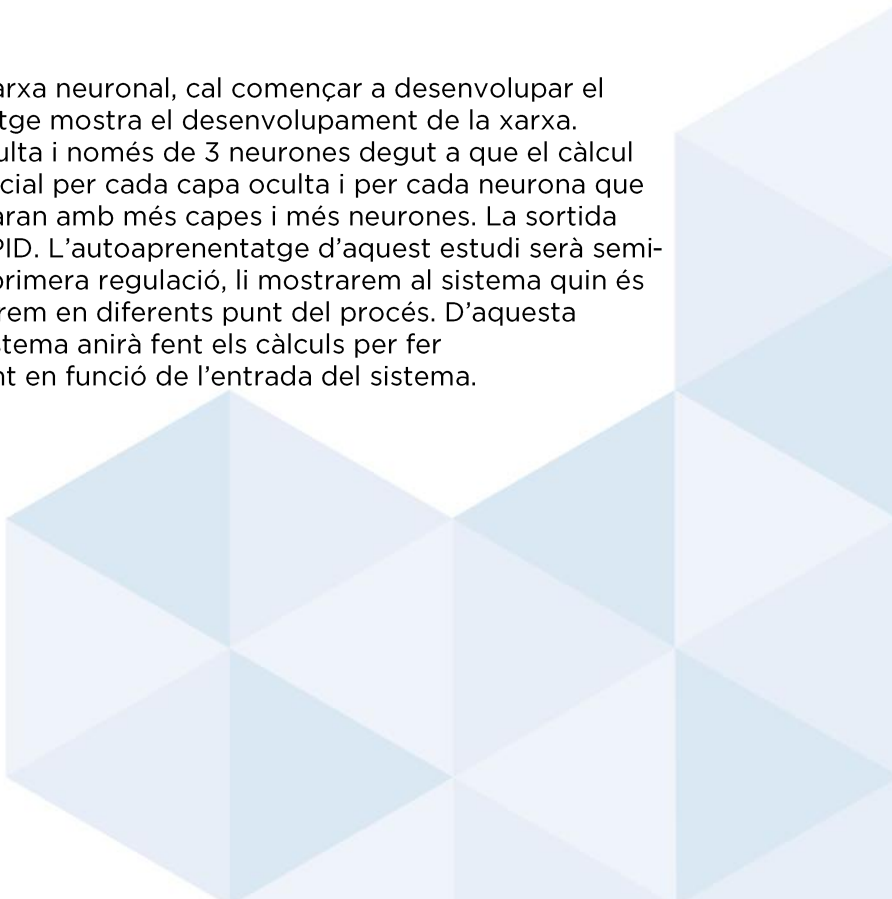
Error derivatiu

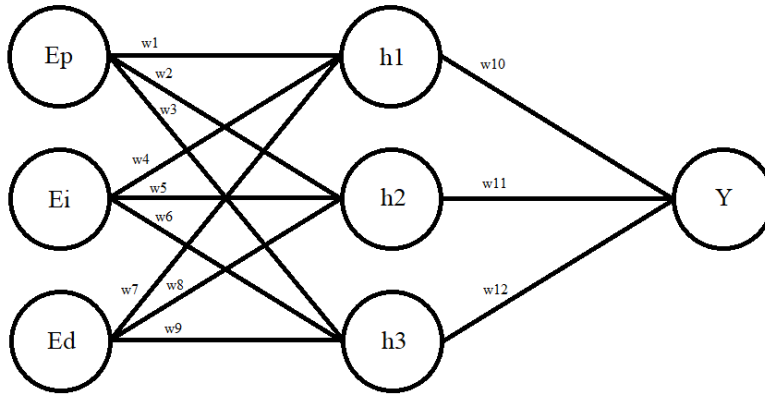
L'error derivatiu és la pendent que genera l'error proporcional en funció del temps. Nosaltres calcularem aquesta pendent restant l'error proporcional actual, el valor de l'error proporcional és la mitjana de la diferència d'error dels últims 5 segons.

$$u(t) = \frac{de(t)}{dt}$$

$$Ed = \frac{Ep_{act} - Ep_{-4}}{5}$$

Tenint ja les dades d'entrada a la xarxa neuronal, cal començar a desenvolupar el càlcul de la sortida. La següent imatge mostra el desenvolupament de la xarxa. Aquesta només tindrà una capa oculta i només de 3 neurones degut a que el càlcul de la xarxa creix de forma exponencial per cada capa oculta i per cada neurona que s'afegeix. Les properes proves es faran amb més capes i més neurones. La sortida serà el percentatge de sortida del PID. L'autoaprenentatge d'aquest estudi serà semi-manual. Això vol dir que durant la primera regulació, li mostrarem al sistema quin és el percentatge de sortida que esperem en diferents punt del procés. D'aquesta manera durant aquest procés, el sistema anirà fent els càlculs per fer l'autoaprenentatge de cada moment en funció de l'entrada del sistema.





Per obtenir el càlcul de la sortida en funció de l'entrada, primer hem de calcular els passos entremitjos h1, h2 i h3. Es calcula de la següent manera:

$$h1 = Ep * w1 + Ei * w4 + Ed * w7$$

$$h2 = Ep * w2 + Ei * w5 + Ed * w8$$

$$h3 = Ep * w3 + Ei * w6 + Ed * w9$$

De la mateixa manera obtindrem la sortida Y.

$$Y = h1 * w10 + h2 * w11 + h3 * w12$$

El valor de les variables w1-12 serà definit a l'inici del càlcul amb un valor aleatori d'entre 0 i 1 per a donar un número aleatori a la sortida.

L'error de la sortida Y (predicció) en funció del valor esperat (actual) es calcularà amb la funció de l'error mig quadràtic. Amb la següent fórmula:

$$Error = \frac{1}{2} (predicció - actual)^2$$

La retropropagació, ens ajuda a modificar les variables w_x per adequar-les a l'error de la sortida a partir del descens del gradient i fer així que la xarxa neuronal evolucioni i aprengui de l'error de la sortida.

El descens del gradient és un algoritme per a trobar el mínim en una funció i això ens permet minimitzar l'error que tenim a la sortida. Per trobar el mínim d'una funció utilitzarem la fórmula:

$$*w_x = w_x - a \left(\frac{\partial Error}{\partial w_x} \right)$$

Per exemple, al actualitzar w_{12} , prenem la branca w_{12} i restem la derivada parcial de l'error en funció de w_{12} . Aquesta estarà multiplicar per la variable a (tassa

d'aprenentatge) Aquesta variable farà que el procés d'aprenentatge sigui més ràpid o més lent.

$$*w_{12} = w_{12} - a \left(\frac{\partial Error}{\partial w_{12}} \right)$$

La derivació de la funció de l'error es calcula fent servir la regla de la cadena d'aquesta manera:

$$\frac{\partial Error}{\partial w_{12}} = \frac{\partial \frac{1}{2} (predicció - actual)^2}{\partial predicció} * \frac{\partial predicció}{\partial w_{12}}$$

$$\frac{\partial Error}{\partial w_{12}} = (predicció - actual) * \frac{\partial h1 * w_{10} + h2 * w_{11} + h3 * w_{12}}{\partial w_{12}}$$

$$\frac{\partial Error}{\partial w_{12}} = (predicció - actual) * h3$$

De la mateixa manera trobarem els valors de w_{11} i w_{10}

$$*w_{12} = w_{12} - a(predicció - actual) * h3$$

$$*w_{11} = w_{11} - a(predicció - actual) * h2$$

$$*w_{10} = w_{10} - a(predicció - actual) * h1$$

Però, al retrocedir per actualitzar w_{1-9} la regla de la cadena quedarà així:

$$\frac{\partial Error}{\partial w_9} = \frac{\partial \frac{1}{2} (predicció - actual)^2}{\partial predicció} * \frac{\partial predicció}{\partial h3} * \frac{\partial h3}{\partial w_9}$$

$$\frac{\partial Error}{\partial w_9} = (predicció - actual) * w_{12} * Ed$$

$$*w_9 = w_9 - a(predicció - actual) * w_{12} * Ed$$

$$*w_8 = w_8 - a(predicció - actual) * w_{11} * Ed$$

$$*w_7 = w_7 - a(predicció - actual) * w_{10} * Ed$$

$$*w_6 = w_6 - a(predicció - actual) * w_{12} * Ei$$

$$*w_5 = w_5 - a(predicció - actual) * w_{11} * Ei$$

$$*w_4 = w_4 - a(predicció - actual) * w_{10} * Ei$$

$$*w_3 = w_3 - a(predicció - actual) * w_{12} * Ep$$

$$*w_2 = w_2 - a(predicció - actual) * w_{11} * Ep$$

$$*w_1 = w_1 - a(predicció - actual) * w_{10} * Ep$$

A **ACE Automatisme i Control Elèctric** comptem amb un **equip tècnic**, format per professionals qualificats i en formació contínua, que soluciona els requeriments dels projectes dels nostres clients donant-los suport en el disseny, programació, integració i supervisió dels mateixos. Un servei tècnic d'assessoria i consultoria durant tot el procés de l'oferta.