Lineamientos para la evaluación del curso Inteligencia Artificial Supervisados

O. L. Quintero

25 de abril de 2024

Resumen

Este documento contiene los lineamientos para la presentación de la actividad evaluativa.

1. Introducción

El objetivo del curso es proveer elementos teóricos y conceptuales que le permitan a los estudiantes de Inteligencia Artificial de Ingenieria Matemàtica, enfrentar el problema de construir un modelo compacto (learning machine) que permita representar fenómenos del mundo real. El curso pretende adelantar la correcta aplicación de conceptos que permiten construir el modelo, evaluarlo y juzgar con perspectiva científica su desempeño.

En esta actividad vamos a dar continuidad al ejercicio DE la clase, ESTA VEZ VAMOS A APRENDER COMO SE PUEDE REALIZAR LA RETRO-PROPAGACION de los errores de salida mediante un gradiente local calculado de la siguiente manera:

$$\begin{aligned}
\delta_j(n) &= -\frac{\delta \mathscr{E}(n)}{\delta v_j(n)} \\
&= -\frac{\delta \mathscr{E}(n)}{\delta e_j(n)} \frac{\delta e_j(n)}{\delta y_j(n)} \frac{\delta y_j(n)}{\delta v_j(n)} \\
&= \quad e_j(n) \varphi_j'(v_j(n))
\end{aligned} \tag{1}$$

Esta reformulacion de gradiente, nos permite plantear una regla de aprendizaje para la neurona oculta j igual a la que habiamos realizado con la neurona de salida:

- efecto del error en la neurona $e_j(n)$ pero hay error en j??
- Variacion de la funcion de activacion $\varphi_j'(v_j(n))$ en la neurona j
- y la entrada a la neurona j (viniendo de la entrada i, $y_i(n)$).

2. Calculando la regla de aprendizaje para la neurona oculta j

No tenemos error que medir en la neurona oculta como les habia dicho en clase. Tenemos la energia instantanea del error que vimos en clase:

$$\mathscr{E}(n) = \frac{1}{2}\sum_{k\in C} e_k^2(n) \tag{2}$$

las neuronas de salida k tienen entradas que vienen de j y no hay error en $j$que podamos usar para calcular los pesos $w_{ji}(n)$.

$$\begin{aligned} e_k(n) =& d_k(n) - y_k(n) \\ =& d_k(n) - \varphi_k(v_k(n)) \end{aligned} \tag{3}$$

Reexpresando la derivada interna respecto a la salida j en la ecuacion 12 nos saltamos el termino del error,

$$\begin{aligned} \delta_j(n) =& -\frac{\delta\mathscr{E}(n)}{\delta y_j(n)}\frac{\delta y_j(n)}{\delta v_j(n)} \\ =& -\frac{\delta\mathscr{E}(n)}{\delta y_j(n)}\varphi_j'(v_j(n)) \end{aligned} \tag{4}$$

Necesitamos encotrar la nueva expresion para el gradiente local en la neurona j

$$\frac{\delta\mathscr{E}(n)}{\delta y_j(n)} = \sum_k e_k \frac{\delta e_k(n)}{\delta y_j(n)} \tag{5}$$

Para llegar a $\delta y_j(n)$ recuerden que $y_j(n)$ es la entrada a la neurona k,

$$\frac{\delta\mathscr{E}(n)}{\delta y_j(n)} = \sum_k e_k(n) \frac{\delta e_k(n)}{\delta v_k(n)}\frac{\delta v_k(n)}{\delta y_j(n)} \tag{6}$$

Desarrollando,

$$\frac{\delta e_k(n)}{\delta v_k(n)} = -\varphi_k'(v_k(n)) \tag{7}$$

Recordando para l neuronas de la capa oculta j, el campo local inducido de la neurona k se escribe asi:

$$v_k(n) = \sum_{j=0}^{l} w_{kj}(n) y_j(n) \tag{8}$$

Reescribiendo,

$$\frac{\delta v_k(n)}{\delta y_j(n)} = w_{kj}(n) \tag{9}$$

Finalmente:

$$
\begin{aligned}
\frac{\delta \mathscr{E}(n)}{\delta y_j(n)} &= -\sum_k e_k(n)\varphi'_k(v_k(n))w_{kj}(n) \\
&= -\sum_k \delta_k(n) w_{kj}(n)
\end{aligned}
\tag{10}
$$

Nueva expresion para j es:

$$
\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \tag{11}
$$

Debe usar para la capa de salida, la nueva expresion para el gradiente local asi:

$$
\delta_k(n) = \quad e_k(n)\varphi'_k(v_k(n)) \tag{12}
$$

3. Tarea

El estudiante debe USAR LOS DATOS QUE LES VOY A DAR y realizar este ejercicio de manera INDIVIDUAL. Elegir del espacio de 5 dimensiones, las entradas y las salidas que desee, siempre y cuando cumpla con el criterio elegido de entrenamiento en la proxima seccion.

El estudiante puede usar cualquier tipo de programa para realizar el ejercicio que se detallará en la siguiente sección.

3.1. Entrenamiento de la red neuronal

Para su problema con n entradas, m salidas y N cantidad de instancias (patrones o tamaño de la muestra).

Seleccionar una red neuronal en configuracion perceptron multicapa con n entradas, m salidas.

Realizar el entrenamiento del perceptron multicapa con la combinatoria de arquitecturas de la siguiente manera:

- Con L capas ocultas, donde L = 1,...,3.

- Variando el numero de neuronas de las capas ocultas l_i = 1,...,5

De esta manera, usted verificará el proceso de aprendizaje de la red neuronal a medida que va aumentando el número de pesos que debe estimar, usando el mismo tamaño de la muestra N.

Como podra observar, lo que se solicita es que entrene perceptrones multicapa de menor a mayor tamaño y profundidad desde una red mínima (n entradas, 1 neurona en la capa oculta y m salidas), hasta una red de gran tamaño (n entradas, 3 capas ocultas, 5 neuronas en la capa oculta y m salidas).

Este entrenamiento se realizará variando la tasa de aprendizaje en tres valores: 0.2, 0.5 y 0.9.

Use su modo secuencial obligatoriamente o en lotes cuando quiera y considere apropiado. Pero me interesa que trabaje secuencial.

El entrenamiento debe realizarse usando un criterio de parada de un cambio de gradiente local de la capa de salida que cambie poco dentro de una banda de alrededor del 0.01 y numero maximo de epocas de entrenamiento de 50. Si los gradientes se detienen puede detener el entrenamiento.

Debe observar las curvas de entrenamiento y de aprendizaje de cada una de las redes mediante la observacion de las energias de error y los gradientes me interesan mas los gradientes

3.2. Entregables

Se debe entregar:

- Codigos elaborados COMENTADOS APROPIADAMENTE