Algoritmo de Backpropagation

Backpropagation algorithm

Autor 1: David Arcila Parra

Auror 2: Liz Dahianna Noreña Giraldo

*Computación Blanda, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: [davidnon@utp.edu.co](mailto:davidnon@utp.edu.co).

[lizgiraldo@utp.edu.co](mailto:lizgiraldo@utp.edu.co)

**Resumen—***En redes neuronales se busca ajustar los pesos de cada neurona de tal manera que se minimice el error. El algoritmo de backpropagation nos indica cuánto de culpa tiene cada neurona del error global cometido.*

*La forma en que como se calcula la culpa que tiene cada neurona en el error es lo que da sentido al nombre de backpropagation, ya que primeramente calcula la culpa del error de cada neurona de la última capa y lo va propagando hacia atrás para ver cuánta culpa tienen el resto.*

*Se podría decir que pondera el reparto del error para cada una de las neuronas de la red.*

*El algoritmo de backpropagation determina la culpa del error, calculando las derivadas parciales de la función de coste con respecto a cada una de las variables*

***Palabras clave---- Algoritmo, Capa. Neurona, Backpropagation, Error, Variable.***

***Abstract—*** *In neural networks, we seek to adjust the weights of each neuron in such a way that the error is minimized. The backpropagation algorithm tells us how much fault each neuron has of the global error committed.*

*The way in which the fault that each neuron has in the error is calculated is what gives meaning to the name of backpropagation, since it first calculates the fault of the error of each neuron of the last layer and propagates it back to see how much It's the fault of the rest.*

*You could say that weights the distribution of error for each of the neurons in the network.*

*The backpropagation algorithm determines the fault of the error, calculating the partial derivatives of the cost function with respect to each of the variables*

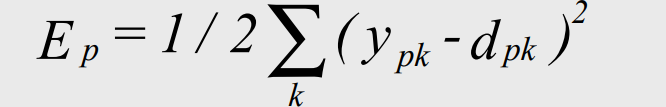
***Key Word — algorithm, layer. Neuronal, Backpropagation, Error, Variable.***

1. INTRODUCCIÓN

El algoritmo de Backpropagation es utilizado para entrenar redes multicapa

Exige que la función de activación de las neuronas sea derivable y creciente. Las funciones comúnmente escogidas son: sigmoide, logística, tangente hiperbólica.

Pretende minimizar el error cuyo dominio es el espacio de los pesos de las conexiones, descrito por la siguiente ecuación:



Las redes neuronales artificiales proporcionan un modelo de cómputo paralelo y distribuido capaz de aprender a partir de ejemplos (datos) Los algoritmos de aprendizaje (asociados a modelos concretos de redes) permiten ir modificando los pesos de las conexiones sinápticas de forma que la red aprenda a partir de los ejemplos que se le presentan.[1]

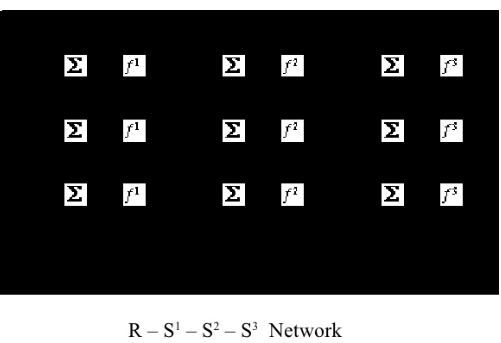
No se programan, se entrenan. Necesitan disponer de ejemplos, en un número suficiente y una distribución representativa para ser capaces de generalizar correctamente. Requieren un proceso de validación para evaluar la “calidad” del aprendizaje conseguido.[1]

1. CONTENIDO
2. **Propiedades de Backpropagation**

* La función de error define una superficie en el espacio de pesos, y estos son modificados sobre el gradiente de la superficie
* Un mínimo local puede existir en la superficie de decisión: esto significa que no hay teorema de convergencia para la retropropagación (el espacio de pesos es lo suficientemente grande que esto rara vez sucede) [2]
* Las redes toman un periodo grande de entrenamiento y muchos ejemplos
* Además mientras la red generaliza, el sobre entrenamiento puede generar un problema.

1. **Arquitectura**

Perceptrón Multicapa



1. **Algoritmo de aprendizaje**

* Los pesos se ajustan después de ver los pares entrada/salida del conjunto de entrenamiento.
* Los pesos se ajustan después de ver los pares entrada/salida del conjunto de entrenamiento.
* En la retropropagación, la salida actual es comparada con la salida deseada, entonces se calcula el error para las unidades de salida
* Entonces los pesos en la salida son ajustados para reducir el error, esto da un error estimado para las neuronas ocultas y así sucesivamente.
* Una época se define como el ajuste de los pesos para todos los pares de entrenamientos, usualmente la red requiere muchas épocas para su entrenamiento.

1. **La variación momento en Backpropagation**

Después que el error es calculado sobre cada conjunto deentrenamiento (patrón), cada peso es ajustado en proporción al gradiente del error calculado propagado hacia atrás desde la salida hacia la entrada, tal como se vio en el algoritmo y el ejemplo. Los cambios en los pesos reducen el error total de la red neuronal. Para evitar que en la superficie del error los pesos estimen una minimización del error con un óptimo local, se aplica el momento como un mecanismo alternativo para cambiar la dirección de los pesoso y eventualmente logre un mínimo global. En consecuencia, mejora los tiempos de entrenamiento al algoritmo backpropagation. En esta caso al aplicar momento, en lugar de calcular los pesos para un conjunto de entrenamiento mediante ∆pwji = ηδ pjZpi , se afecta esta expresión mediante un término que indica la contribución proporcional (término momento) α ⋅ de los pesos previos o anteriores. Es decir, (anterior) ji pw pi Z ji pj ∆pw =ηδ +α ⋅∆ , el valor de α ⋅debe estar entre 0 y 1. Con respecto a esta variación, se ha encontrado que en algunos casos puede producir buenos resultados cuando en la superficie del error alcanza un área plana o un mínimo local puntual (profundidad estrecha y bastante grande). En estos casos valores cercano a uno (por ej.: 0.9) podrían ayudar a mejorar la solución (permiten obviar este mínimo local). En otros casos, en la superficie del error, por ejemplo áreas ondulantes amplias y pocas profundas, no ayudarían en mayor grado. De ahí que la solución pueda estar afectada por varios mínimos locales, haciendo inalcanzable un mínimo global. [3]

En backpropagation, el método general de entrenamiento se resume a cuatro pasos: Pasos hacia delante:

1. Selecciona un vector de entrada desde el conjunto de entrenamiento.

2. Aplica esta entrada a la red y calcula la salida. Análisis Multivariante

3. Calcular el error entre la salida calculada y la salida deseada de la entrada usada.

4. Ajustar los pesos para que el error cometido entre la salida calculada y la salida deseada sea disminuido.

5. Repetir los pasos 1 al 5 para todas las entradas del conjunto de entrenamiento, hasta que el error global sea aceptablemente bajo.

### Aplicaciones de las redes backpropagation

Este tipo de redes se están aplicando a distintas clases de problemas. Esta versatilidad se debe a la naturaleza general de su proceso de aprendizaje ya que solamente se necesitan dos ecuaciones para propagar las señales de error hacia atrás. La utilización de una u otra ecuación sólo depende de si la unidad de proceso es o no de salida. Algunos de los campos de aplicación más representativos son :

* *Codificación de información* : la idea consiste en que la información de entrada se recupere en la salida a través de un código interno.
* *Traducción de texto en lenguaje hablado*
* *Reconocimiento de lenguaje hablado*
* *Reconocimiento óptico de caracteres (OCR)*
* *Aplicaciones en cardiología*
  + Clasificación de señales electrocardiográficas (ECG)
  + Detección de taquicardias ventriculares y supraventriculares
  + Detección de complejos QRS anómalos
  + Reconocimiento de formas anormales en señales ECG
  + Emulación hardware de una red neuronal para el procesado de ECG
  + Cancelación de ruido en señales ECG
* *Comprensión/descomprensión de datos*

1. CONCLUSIONES

* Al escoger las redes neuronales como método de solución de un problema en particular, es necesario tener un entendimiento profundo y completo de lo que es el problema como tal, pues ello facilita la elección de los patrones de entrenamiento y ofrece una idea general de la arquitectura que debe tener la red y de lo que se espera de ella.
* El algoritmo de retropropagación normalmente converge razonablemente rápido. Sin embargo, la velocidad real depende mucho de los parámetros de simulación en los valores de peso iniciales.

REFERENCIAS

[1] Backpropagation, Dsiponible en: <https://elvex.ugr.es/decsai/computational-intelligence/slides/N2%20Backpropagation.pdf>

[2] Algoritmo de Backpropagation, Disponible en: <https://es.slideshare.net/mentelibre/algoritmo-de-retropropagacin>

[3] Backpropagation Disponible en: <http://webdelprofesor.ula.ve/economia/gcolmen/programa/economia/backpropagation_rna.pdf>