Redes Neuronales

Sesión 03 - Módulo Reto

Alejandro Ucan

2024-11-20

Objetivos

- Introducir las bases de *redes neuronales*.
- Introducir las funciones de activación.
- Introducir el concepto de *backpropagation*.

Motivación

- Las redes neuronales son un modelo computacional que se inspira en la estructura y funcionamiento del cerebro humano.

- Las redes neuronales son capaces de aprender basados en datos y realizar tareas como clasificación, regresión, segmentación, entre otras.

Redes Neuronales

> Las **redes neuronales**, también conocidas como *Artificial Neural Networks* (ANN), son sistemas computacionales que modelan una relación entre un conjunto de señales de entrada y un conjunto de señales de salida.
>
> Estas redes están conformadas por *neuronas* que se relacionan por medio de *sinapsis*.

Neuronas (de a de veras)

Biológicamente, una neurona es una célula que se encarga de transmitir información.

Esta recibe señales eléctricas por medio de sus dendritas.

La acumulación de estas señales hace que se active una respuesta (basado en un umbral) dado por el axón.

La señal se transmite a otras neuronas vecinas.

Neurona Artificial

Recibe información de entrada $x_1, x_2, \ldots, x_n$ o señales.

Procesa la información y la "acumula" hasta que la función de *activación* llega al umbral deseado.

La neurona artificial emite una señal de salida y, y en caso dado de conexión la transmite a otras neuronas.

Modelo de Neurona Artificial

Dado lo anterior, la expresión que relaciona las entradas y la salida de una neurona artificial es:

$$y = \phi\left(\sum_{i=0}^{n} w_i x_i\right) = \phi(w^T x)$$

donde $w \in \mathbb{R}^{n+1}$ es un vector de pesos, $x_0 = 1$ y ϕ es la *función de activación*.

Funciones de Activición

La función de activación es el mecanismo por medio del cual la neurona procesa la información de entrada y genera una señal de salida. Las función de activación más comunes son:

- **Sigmoide:** $\sigma(x) = \frac{1}{1+e^{-x}}$
- **Tangente hiperbólica:** $\phi(x) = 2\sigma(x) - 1$
- **Unidad Lineal Rectificada (ReLU):** $\phi(x) = \max(0, x)$
- **Identidad:** $\phi(x) = x$

Backpropagation

El entrenamiento de una red neuronal que resuelve un problema de clasificación se realiza minimizando la siguente función de costo:

$$C(W) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\left(y_k^{(i)}\log(h_w(x^{(i)}))_k + (1-y_k^{(i)})\log(1-h_w(x^{(i)}))_k\right)$$

donde K es el número de neuronas en la capa de salida.

Backpropagation

Entrenar una red neuronal significa: encontrar los pesos W óptimos que minimizan la función de costo anterior. Para ello necesitamos el algoritmo de *descenso gradiente* y es posible gracias al algoritmo de *backpropagation*.

Esto es, **backpropagation** nos permite calcular estas derivadas parciales

$$\frac{\partial C}{\partial w_{ij}^{(l)}}$$

la derivada parcial respecto a la entrada j de la neurona i de la capa $l+1$.

Backpropagation

Sea $a^{(l)}$ con $l = 2, \cdots, L$ es un vector que contiene las *activaciones* de las neuronas de la capa l, con $a^{(1)} = x$.

Sea $\delta^{(L)} = a^{(L)} - y$ el error entre la salida esperada (y) y lo entregado por la última capa de la red neuronal $a^{(L)}$.

Definimos $\delta^{(l)}$, con $l = 2 \cdots, L-1$ como

$$\delta^{(l)} = ((W^{(l)})^T \delta^{(l+1)}) \odot a^{(l)} \odot (1 - a^{(l)})$$

donde $\odot$ es el producto de Hadamard.

Backpropagation (Pseudo-código)

Dado un conjunto de entrenamiento $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$,$ el algoritmo de backpropagation es el siguiente:

1. **Inicializar** $\Delta^{(l)} = 0$ para $l = 1, \cdots, L-1, i = 1$
2. **repetir** n veces: $a^{(1)} = x^{(i)}$
   calcular $a^{(l)}$ para $l = 2, \cdots, L$
   $\delta^{(L)} = a^{(L)} - y^{(i)}$
   calcular $\delta^{(l)}$ para $l = L-1, \cdots, 2$
   calcular $\Delta^{(l)} = \Delta^{(l)} + \delta^{(l+1)}(a^{(l)})^T, l = 1, 2, \cdots, L-1$
   $i = i + 1$
3. **retornar** $D^{(l)} = \frac{1}{n}\Delta^{(l)}$ para $l = 1, 2, \cdots, L-1$

Backpropagation (Pseudo-código)

Teniendo en cuenta lo anterior, el entrenamiento de una red neuronal totalmente conectada se puede llevar a cabo ejecutando el siguiente algoritmo:

1. **inicializar** $W = W_0, \alpha \in (0, 1)$
2. **repetir calcular** $\nabla C(W)$ usando el algoritmo de backpropagation
   $W = W - \alpha \nabla C(W)$
3. **hasta** cumplir con un criterio de parada.
4. **retornar** W

Referencias

1. Haykin, S. (2009). Neural Networks and Learning Machines. Pearson Education.