Redes Neuronales Multicapa

Marcos Esteve Casademunt

Enero 2019

1. Ejercicio 1

Uno de los problemas que presenta el algoritmo BackProp es que converge lentamente. Una posible solución pasa por añadir una inercia o momentum con un peso $0 \leq v < 1$.

La implementación resulta muy similar a la del algoritmo original. Tendremos que añadir un parametro de entrada v y módificar $\Delta\Theta$ añadiendo el nuevo parámetro.

1.1. Batch

Entrada: Topología, pesos iniciales $\theta_{ij}^{l} 1 \leq l \leq L, 1 \leq i \leq M_l, 0 \leq j \leq M_{l-1}$, factor de aprendizaje p, condiciones de convergencia, N datos de entrenamiento S, **momentum** $0 \leq v < 1$.

Salidas: Pesos de las conexiones que minimizan el error cuadrático medio de S

Mientras no se cumplan las condiciones de convergencia:
- Para $1 \leq l \leq L, 1 \leq i \leq M_l, 0 \leq j \leq M_{l-1}$, inicializar $\Delta\theta_{ij}^{l} = 0$
- Para cada muestra de entrenamiento (x,t) $\in$ S
  - Desde la capa de entrada a la de salida(l=1 , ... , L)
    - Para $1 \leq i \leq M_l$ calcular $\phi_i^l \; y \; s_i^l = g(\phi_i^l)$
  - Desde la capa de salida a la de entrada (l = L , ..., 1)
    - Para cada nodo $(1 \leq i \leq M_l,)$
      - Calcular $\delta_i^l = g^{'}(\phi_i^l)(t_{ni-s_i^L}) \; si \; l == L \;\; o \;\; g^{'}(\phi_i^l)(\sum_r \delta_r^{l+1}\theta_{ri}^{l+1})$
      - Para cada peso $\theta_{ij}^{l}(0 \leq j \leq M_{l-1}) \; calcular : \Delta\theta_{ij}^{l} = \mathbf{v}\Delta\theta_{ij}^{l} + p\delta_i^l \; s_j^{l-1}$
  - Para $1 \leq l \leq L, 1 \leq i \leq M_l, 0 \leq j \leq M_{l-1}$ Actualizar pesos: $\theta_{ij}^{l} = \theta_{ij}^{l} + \frac{1}{N}\Delta\theta_{ij}^{l}$

1.2. Incremental

Entrada: Topología, pesos iniciales $\theta_{ij}^{l} 1 \leq l \leq L, 1 \leq i \leq M_l, 0 \leq j \leq M_{l-1}$, factor de aprendizaje p, condiciones de convergencia, N datos de entrenamiento S, **momentum** $0 \leq v < 1$.

Salidas: Pesos de las conexiones que minimizan el error cuadrático medio de S

Mientras no se cumplan las condiciones de convergencia:
Para cada muestra de entrenamiento (x,t) $\in$ S
Desde la capa de entrada a la de salida(l=1 , ... , L)
Para $1 \leq i \leq M_l$ calcular $\phi_i^l \ y \ s_i^l = g(\phi_i^l)$
Desde la capa de salida a la de entrada (l = L , ..., 1)
Para cada nodo ($1 \leq i \leq M_l,$)
Calcular $\delta_i^l = g^{'}(\phi_i^l)(t_{ni-s_i^L}) \ si \ l == L \ \ o \ \ g^{'}(\phi_i^l)(\sum_r \delta_r^{l+1}\theta_{ri}^{l+1})$
Para cada peso $\theta_{ij}^l(0 \leq j \leq M_{l-1}) \ actualizar : \Theta_{ij}^l = \ \ \mathbf{v}\Theta_{ij}^l + \frac{p}{N}\delta_i^l s_j^{l-1}$

2. Ejercicio 2

Con el fin de evitar que los pesos sean muy grandes y por tanto provoquen una paralisis de la red se pretende implantar un factor de amortiguamiento en el algoritmo BackProp

La implementación resulta muy similar a la del algoritmo original. Tendremos que añadir un parametro de entrada λ y módificar $\Delta\Theta$ añadiendo el nuevo parámetro.

2.1. Batch

Entrada: Topología, pesos iniciales $\theta_{ij}^l 1 \leq l \leq L, 1 \leq i \leq M_l, 0 \leq j \leq M_{l-1}$, factor de aprendizaje p, condiciones de convergencia, N datos de entrenamiento S. **Factor de regularización** λ.

Salidas: Pesos de las conexiones que minimizan el error cuadrático medio de S

Mientras no se cumplan las condiciones de convergencia:
Para $1 \leq l \leq L, 1 \leq i \leq M_l, 0 \leq j \leq M_{l-1}$, inicializar $\Delta\theta_{ij}^l = 0$
Para cada muestra de entrenamiento (x,t) $\in$ S
Desde la capa de entrada a la de salida(l=1 , ... , L)
Para $1 \leq i \leq M_l$ calcular $\phi_i^l \ y \ s_i^l = g(\phi_i^l)$
Desde la capa de salida a la de entrada (l = L , ..., 1)
Para cada nodo ($1 \leq i \leq M_l,$)
Calcular $\delta_i^l = g^{'}(\phi_i^l)(t_{ni-s_i^L}) \ si \ l == L \ \ o \ \ g^{'}(\phi_i^l)(\sum_r \delta_r^{l+1}\theta_{ri}^{l+1})$
Para cada peso $\theta_{ij}^l(0 \leq j \leq M_{l-1}) \ calcular : \Delta\theta_{ij}^l = p\delta_i^l \ s_j^{l-1} \ -2 \ p \ \lambda \ \theta_{ij}^l$
Para $1 \leq l \leq L, 1 \leq i \leq M_l, 0 \leq j \leq M_{l-1}$ Actualizar pesos: $\theta_{ij}^l = \theta_{ij}^l + \frac{1}{N}\Delta\theta_{ij}^l$

2.1.1. Incremental

Entrada: Topología, pesos iniciales $\theta_{ij}^{l} 1 \leq l \leq L, 1 \leq i \leq M_l, 0 \leq j \leq M_{l-1}$, factor de aprendizaje p, condiciones de convergencia, N datos de entrenamiento S,**Factor de regularización** λ.

Salidas: Pesos de las conexiones que minimizan el error cuadrático medio de S

Mientras no se cumplan las condiciones de convergencia:

Para cada muestra de entrenamiento (x,t) $\in$ S

Desde la capa de entrada a la de salida(l=1 , ... , L)

Para $1 \leq i \leq M_l$ calcular $\phi_i^l \; y \; s_i^l = g(\phi_i^l)$

Desde la capa de salida a la de entrada (l = L , ..., 1)

Para cada nodo $(1 \leq i \leq M_l,)$

Calcular $\delta_i^l = g^{'}(\phi_i^l)(t_{ni - s_i^L}) \; si \; l == L \;\; o \;\; g^{'}(\phi_i^l)(\sum_r \delta_r^{l+1} \theta_{ri}^{l+1})$

Para cada peso $\theta_{ij}^l (0 \leq j \leq M_{l-1}) \; actualizar : \Theta_{ij}^l = -2 \; p \; \lambda \Theta_{ij}^l + \frac{p}{N} \delta_i^l s_j^{l-1}$