Mejorando la Generalización

Fernando Lozano

Universidad de los Andes

2 de septiembre de 2014

Regularización y complejidad del modelo

Regularización y complejidad del modelo

- Existe un compromiso entre la complejidad de las funciones que se pueden implementar con una red neuronal y el ajuste a los datos de entrenamiento.

Regularización y complejidad del modelo

- Existe un compromiso entre la complejidad de las funciones que se pueden implementar con una red neuronal y el ajuste a los datos de entrenamiento.
  - Una red que implemente funciones muy complejas, puede sobre ajustarse a los datos, y resultar en pobre generalización.

Regularización y complejidad del modelo

- Existe un compromiso entre la complejidad de las funciones que se pueden implementar con una red neuronal y el ajuste a los datos de entrenamiento.
  - Una red que implemente funciones muy complejas, puede sobre ajustarse a los datos, y resultar en pobre generalización.
  - Una red muy simple puede no ser suficiente para capturar la estructura en los datos.

Regularización y complejidad del modelo

- Existe un compromiso entre la complejidad de las funciones que se pueden implementar con una red neuronal y el ajuste a los datos de entrenamiento.
    - Una red que implemente funciones muy complejas, puede sobre ajustarse a los datos, y resultar en pobre generalización.
    - Una red muy simple puede no ser suficiente para capturar la estructura en los datos.
- El problema con redes muy complejas es que pueden existir muchas redes que se ajusten bien a los datos.

Regularización y complejidad del modelo

- Existe un compromiso entre la complejidad de las funciones que se pueden implementar con una red neuronal y el ajuste a los datos de entrenamiento.
  - Una red que implemente funciones muy complejas, puede sobre ajustarse a los datos, y resultar en pobre generalización.
  - Una red muy simple puede no ser suficiente para capturar la estructura en los datos.
- El problema con redes muy complejas es que pueden existir muchas redes que se ajusten bien a los datos.
- Esto implica que el modelo que se obtiene al entrenar la red puede tener alta varianza.

Regularización y complejidad del modelo

- Existe un compromiso entre la complejidad de las funciones que se pueden implementar con una red neuronal y el ajuste a los datos de entrenamiento.
  - Una red que implemente funciones muy complejas, puede sobre ajustarse a los datos, y resultar en pobre generalización.
  - Una red muy simple puede no ser suficiente para capturar la estructura en los datos.
- El problema con redes muy complejas es que pueden existir muchas redes que se ajusten bien a los datos.
- Esto implica que el modelo que se obtiene al entrenar la red puede tener alta varianza.
- Diferentes técnicas tratan de balancear este compromiso.

El dilema sesgo-varianza

- Análisis para el problema de regresión.

El dilema sesgo-varianza

- Análisis para el problema de regresión.
- Suponga que $h \in \mathcal{H}$ depende de parámetros $\mathbf{w}$.

El dilema sesgo-varianza

- Análisis para el problema de regresión.
- Suponga que $h \in \mathcal{H}$ depende de parámetros $\mathbf{w}$.
- El error cuadrático:

$$\hat{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(h(\mathbf{x}_i, \mathbf{w}) - y_i)^2$$

El dilema sesgo-varianza

- Análisis para el problema de regresión.
- Suponga que $h \in \mathcal{H}$ depende de parámetros $\mathbf{w}$.
- El error cuadrático:

$$\hat{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(h(\mathbf{x}_i, \mathbf{w}) - y_i)^2$$

- En el límite $n \to \infty$:

$$E(\mathbf{w}) = \lim_{n\to\infty} \frac{\hat{E}(\mathbf{w})}{n}$$

El dilema sesgo-varianza

- Análisis para el problema de regresión.
- Suponga que $h \in \mathcal{H}$ depende de parámetros $\mathbf{w}$.
- El error cuadrático:

$$\hat{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(h(\mathbf{x}_i, \mathbf{w}) - y_i)^2$$

- En el límite $n \to \infty$:

$$E(\mathbf{w}) = \lim_{n\to\infty} \frac{\hat{E}(\mathbf{w})}{n} = \frac{1}{2}\iint (h(\mathbf{x}, \mathbf{w}) - y)^2 p(\mathbf{x}, y) dy d\mathbf{x}$$

El dilema sesgo-varianza

- Análisis para el problema de regresión.
- Suponga que $h \in \mathcal{H}$ depende de parámetros $\mathbf{w}$.
- El error cuadrático:

$$\hat{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(h(\mathbf{x}_i, \mathbf{w}) - y_i)^2$$

- En el límite $n \to \infty$:

$$E(\mathbf{w}) = \lim_{n\to\infty} \frac{\hat{E}(\mathbf{w})}{n} = \frac{1}{2}\iint (h(\mathbf{x}, \mathbf{w}) - y)^2 p(\mathbf{x}, y) dy d\mathbf{x}$$

- Reemplazando $p(\mathbf{x}, y) = p(y|\mathbf{x})p(\mathbf{x})$:

$$E(\mathbf{w}) = \frac{1}{2}\iint (h(\mathbf{x}, \mathbf{w}) - y)^2 p(y|\mathbf{x})p(\mathbf{x}) dy d\mathbf{x}$$

- Truco: $(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}) + \mathbb{E}(y|\mathbf{x}) - y)^2$

- Truco: $(h(\mathbf{x},\mathbf{w}) - y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}) + \mathbb{E}(y|\mathbf{x}) - y)^2$

$$
\begin{aligned}
(h(\mathbf{x},\mathbf{w}) - y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 + (\mathbb{E}(y|\mathbf{x}) - y)^2 \\
+ 2(h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)
\end{aligned}
$$

- Truco: $(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) \color{red}{- \mathbb{E}(y|\mathbf{x}) + \mathbb{E}(y|\mathbf{x})} - y)^2$

$$\begin{aligned}(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 + (\mathbb{E}(y|\mathbf{x}) - y)^2 \\ + 2(h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)\end{aligned}$$

- integrando cada término con respecto a y:

- Truco: $(h(\mathbf{x},\mathbf{w}) - y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}) + \mathbb{E}(y|\mathbf{x}) - y)^2$

$$(h(\mathbf{x},\mathbf{w}) - y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 + (\mathbb{E}(y|\mathbf{x}) - y)^2 \\ + 2(h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)$$

- integrando cada término con respecto a y:

$$\iint (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(y|\mathbf{x})p(\mathbf{x})dyd\mathbf{x}$$

$$=$$

- Truco: $(h(\mathbf{x},\mathbf{w}) - y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}) + \mathbb{E}(y|\mathbf{x}) - y)^2$

$$\begin{aligned}(h(\mathbf{x},\mathbf{w}) - y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 + (\mathbb{E}(y|\mathbf{x}) - y)^2 \\ + 2(h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)\end{aligned}$$

- integrando cada término con respecto a y:

$$\begin{aligned}&\iint (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(y|\mathbf{x}) p(\mathbf{x}) dy d\mathbf{x} \\ &\qquad = \int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x}\end{aligned}$$

- Truco: $(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}) + \mathbb{E}(y|\mathbf{x}) - y)^2$

$$
\begin{aligned}
(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 + (\mathbb{E}(y|\mathbf{x}) - y)^2 \\
+ 2(h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)
\end{aligned}
$$

- integrando cada término con respecto a y:

$$
\begin{aligned}
\iint (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(y|\mathbf{x}) p(\mathbf{x}) dy d\mathbf{x} \\
= \int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x}
\end{aligned}
$$

$$
\int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y) p(y|\mathbf{x}) dy =
$$

- Truco: $(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}) + \mathbb{E}(y|\mathbf{x}) - y)^2$

$$\begin{aligned}(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 + (\mathbb{E}(y|\mathbf{x}) - y)^2 \\ + 2(h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)\end{aligned}$$

- integrando cada término con respecto a y:

$$\begin{aligned}\iint (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(y|\mathbf{x})p(\mathbf{x})dyd\mathbf{x} \\ = \int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x})d\mathbf{x}\end{aligned}$$

$$\int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)p(y|\mathbf{x})dy = 0$$

- Truco: $(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}) + \mathbb{E}(y|\mathbf{x}) - y)^2$

$$\begin{aligned}(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 + (\mathbb{E}(y|\mathbf{x}) - y)^2 \\ + 2(h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)\end{aligned}$$

- integrando cada término con respecto a y:

$$\begin{aligned}\iint (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(y|\mathbf{x})p(\mathbf{x})dyd\mathbf{x} \\ = \int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x})d\mathbf{x}\end{aligned}$$

$$(h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - \int y)p(y|\mathbf{x})dy = 0$$

- Truco: $(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}) + \mathbb{E}(y|\mathbf{x}) - y)^2$

$$\begin{aligned}(h(\mathbf{x},\mathbf{w})-y)^2 = (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 + (\mathbb{E}(y|\mathbf{x}) - y)^2 \\ + 2(h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)\end{aligned}$$

- integrando cada término con respecto a y:

$$\begin{aligned}\iint (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(y|\mathbf{x})p(\mathbf{x})dyd\mathbf{x} \\ = \int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x})d\mathbf{x}\end{aligned}$$

$$\int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))(\mathbb{E}(y|\mathbf{x}) - y)p(y|\mathbf{x})dy = 0$$

$$\int (\mathbb{E}(y|\mathbf{x}) - y)^2 p(y|\mathbf{x}) dy =$$

$$\int (\mathbb{E}(y|\mathbf{x}) - y)^2 p(y|\mathbf{x}) dy = \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2$$

$$\int (\mathbb{E}(y|\mathbf{x}) - y)^2 p(y|\mathbf{x}) dy = \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2$$

- Todo junto:

$$E(\mathbf{w}) = \frac{1}{2} \int (h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} + \frac{1}{2} \int \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2 p(\mathbf{x}) d\mathbf{x}$$

$$\int (\mathbb{E}(y|\mathbf{x}) - y)^2 p(y|\mathbf{x}) dy = \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2$$

- Todo junto:

$$E(\mathbf{w}) = \frac{1}{2}\int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} + \frac{1}{2}\int \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2 p(\mathbf{x}) d\mathbf{x}$$

- Segundo término no depende de **w**!

$$\int (\mathbb{E}(y|\mathbf{x}) - y)^2 p(y|\mathbf{x}) dy = \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2$$

- Todo junto:

$$E(\mathbf{w}) = \frac{1}{2} \int (h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} + \frac{1}{2} \int \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2 p(\mathbf{x}) d\mathbf{x}$$

- Segundo término no depende de $\mathbf{w}$!
- $h(\mathbf{x}, \mathbf{w})$ que minimiza $E(\mathbf{w})$ es

$$\int (\mathbb{E}(y|\mathbf{x}) - y)^2 p(y|\mathbf{x}) dy = \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2$$

- Todo junto:

$$E(\mathbf{w}) = \frac{1}{2}\int (h(\mathbf{x},\mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} + \frac{1}{2}\int \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2 p(\mathbf{x}) d\mathbf{x}$$

- Segundo término no depende de $\mathbf{w}$!
- $h(\mathbf{x},\mathbf{w})$ que minimiza $E(\mathbf{w})$ es $\mathbb{E}(y|\mathbf{x})$.

$$\int (\mathbb{E}(y|\mathbf{x}) - y)^2 p(y|\mathbf{x}) dy = \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2$$

- Todo junto:

$$E(\mathbf{w}) = \frac{1}{2} \int (h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} + \frac{1}{2} \int \mathbb{E}(y^2|\mathbf{x}) - \mathbb{E}(y|\mathbf{x})^2 p(\mathbf{x}) d\mathbf{x}$$

- Segundo término no depende de $\mathbf{w}$!
- $h(\mathbf{x}, \mathbf{w})$ que minimiza $E(\mathbf{w})$ es $\mathbb{E}(y|\mathbf{x})$.
- $\mathbb{E}(y|\mathbf{x})$ se llama función de regresión.

- En la práctica n es finito. Considere conjuntos con n datos $D_1, D_2, \cdots \sim p(\mathbf{x}, y)$.

- En la práctica n es finito. Considere conjuntos con n datos $D_1, D_2, \cdots \sim p(\mathbf{x}, y)$.
- Para un $\mathbf{x}$ fijo $(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2$ depende del D_i particular.

- En la práctica n es finito. Considere conjuntos con n datos $D_1, D_2, \cdots \sim p(\mathbf{x}, y)$.
- Para un $\mathbf{x}$ fijo $(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2$ depende del D_i particular.
- Promediando sobre todos los D posibles:

- En la práctica n es finito. Considere conjuntos con n datos $D_1, D_2, \cdots \sim p(\mathbf{x}, y)$.
- Para un $\mathbf{x}$ fijo $(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2$ depende del D_i particular.
- Promediando sobre todos los D posibles:

$$\mathbb{E}(E(\mathbf{x})) = \mathbb{E}_D(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2$$

- En la práctica n es finito. Considere conjuntos con n datos $D_1, D_2, \cdots \sim p(\mathbf{x}, y)$.
- Para un $\mathbf{x}$ fijo $(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2$ depende del D_i particular.
- Promediando sobre todos los D posibles:

$$\mathbb{E}(E(\mathbf{x})) = \mathbb{E}_D(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 =$$

$$\mathbb{E}_D\left(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}_D h(\mathbf{x}, \mathbf{w}) + \mathbb{E}_D h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x})\right)^2$$

- En la práctica n es finito. Considere conjuntos con n datos $D_1, D_2, \dots \sim p(\mathbf{x}, y)$.
- Para un $\mathbf{x}$ fijo $(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2$ depende del D_i particular.
- Promediando sobre todos los D posibles:

$$\mathbb{E}(E(\mathbf{x})) = \mathbb{E}_D(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 =$$
$$\mathbb{E}_D\left(h(\mathbf{x}, \mathbf{w}) \color{red}{- \mathbb{E}_D h(\mathbf{x}, \mathbf{w}) + \mathbb{E}_D h(\mathbf{x}, \mathbf{w})} - \mathbb{E}(y|\mathbf{x})\right)^2$$

- De nuevo la integral del término cruzado es cero, y tenemos:

- En la práctica n es finito. Considere conjuntos con n datos $D_1, D_2, \cdots \sim p(\mathbf{x}, y)$.
- Para un $\mathbf{x}$ fijo $(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2$ depende del D_i particular.
- Promediando sobre todos los D posibles:

$$\mathbb{E}(E(\mathbf{x})) = \mathbb{E}_D(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x}))^2 = \\ \mathbb{E}_D\left(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}_D h(\mathbf{x}, \mathbf{w}) + \mathbb{E}_D h(\mathbf{x}, \mathbf{w}) - \mathbb{E}(y|\mathbf{x})\right)^2$$

- De nuevo la integral del término cruzado es cero, y tenemos:

$$\mathbb{E}(E(\mathbf{x})) = \mathbb{E}(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}_D(h(\mathbf{x}, \mathbf{w})))^2 \\ + \left(\mathbb{E}_D(h(\mathbf{x}, \mathbf{w})) - \mathbb{E}(y|\mathbf{x})\right)^2$$

- $(\mathbb{E}_D(h(\mathbf{x}, \mathbf{w})) - \mathbb{E}(y|\mathbf{x}))^2$ es el sesgo de h en $\mathbf{x}$.

- $(\mathbb{E}_D(h(\mathbf{x}, \mathbf{w})) - \mathbb{E}(y|\mathbf{x}))^2$ es el sesgo de h en $\mathbf{x}$.
- $\mathbb{E}(h(\mathbf{x}, \mathbf{w}) - \mathbb{E}_D(h(\mathbf{x}, \mathbf{w})))^2$ es la varianza de h en $\mathbf{x}$.

- $(\mathbb{E}_D(h(\mathbf{x},\mathbf{w})) - \mathbb{E}(y|\mathbf{x}))^2$ es el sesgo de h en $\mathbf{x}$.
- $\mathbb{E}(h(\mathbf{x},\mathbf{w}) - \mathbb{E}_D(h(\mathbf{x},\mathbf{w})))^2$ es la varianza de h en $\mathbf{x}$.
- Integrando sobre $\mathbf{x}$:

$$\text{sesgo}^2 = \frac{1}{2}\int \left(\mathbb{E}_D(h(\mathbf{x},\mathbf{w})) - \mathbb{E}(y|\mathbf{x})\right)^2 p(\mathbf{x})d\mathbf{x}$$

$$\text{varianza} = \frac{1}{2}\int \mathbb{E}(h(\mathbf{x},\mathbf{w}) - \mathbb{E}_D(h(\mathbf{x},\mathbf{w})))^2 p(\mathbf{x})d\mathbf{x}$$

Regularización

- Modificación de la función de error que tenga en cuenta la suavidad de la función que implementa la red:

Regularización

- Modificación de la función de error que tenga en cuenta la suavidad de la función que implementa la red:

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda R(\mathbf{w})$$

Regularización

- Modificación de la función de error que tenga en cuenta la suavidad de la función que implementa la red:

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda R(\mathbf{w})$$

- $R(\mathbf{w})$ penaliza valores de los pesos para los cuales la función resultante no es suave.

Regularización

- Modificación de la función de error que tenga en cuenta la suavidad de la función que implementa la red:

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda R(\mathbf{w})$$

- $R(\mathbf{w})$ penaliza valores de los pesos para los cuales la función resultante no es suave.
- El entrenamiento se realiza minimizando la función de error modificada $\tilde{E}$.

Regularización

- Modificación de la función de error que tenga en cuenta la suavidad de la función que implementa la red:

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda R(\mathbf{w})$$

- $R(\mathbf{w})$ penaliza valores de los pesos para los cuales la función resultante no es suave.
- El entrenamiento se realiza minimizando la función de error modificada $\tilde{E}$.
- De esta manera se logra un balance entre el ajuste a los datos y la suavidad de la función.

Regularización

- Modificación de la función de error que tenga en cuenta la suavidad de la función que implementa la red:

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda R(\mathbf{w})$$

- $R(\mathbf{w})$ penaliza valores de los pesos para los cuales la función resultante no es suave.
- El entrenamiento se realiza minimizando la función de error modificada $\tilde{E}$.
- De esta manera se logra un balance entre el ajuste a los datos y la suavidad de la función.
- $R(\mathbf{w})$ debe ser derivable con respecto a $\mathbf{w}$ y debe poderse calcular eficientemente.

Weight Decay

- Uno de los términos más simples:

Weight Decay

- Uno de los términos más simples:

$$R(\mathbf{w}) = \sum_i w_i^2$$

donde la suma recorre todos los pesos de la red.

Weight Decay

- Uno de los términos más simples:

$$R(\mathbf{w}) = \sum_i w_i^2$$

  donde la suma recorre todos los pesos de la red.
- Ridge regression.

Weight Decay

- Uno de los términos más simples:

$$R(\mathbf{w}) = \sum_i w_i^2$$

  donde la suma recorre todos los pesos de la red.
- Ridge regression.
- Pesos grandes resultan en curvatura grande de la función.

Weight Decay

- Uno de los términos más simples:

$$R(\mathbf{w}) = \sum_i w_i^2$$

  donde la suma recorre todos los pesos de la red.
- Ridge regression.
- Pesos grandes resultan en curvatura grande de la función.
- Pesos pequeños producen funciones aproximadamente lineales.

Weight Decay

- Uno de los términos más simples:

$$R(\mathbf{w}) = \sum_i w_i^2$$

  donde la suma recorre todos los pesos de la red.
- Ridge regression.
- Pesos grandes resultan en curvatura grande de la función.
- Pesos pequeños producen funciones aproximadamente lineales.
- En el MLP, corresponde a región de la sigmoide aproximadamente lineal.

- El gradiente modificado:

$$\nabla \tilde{E} = \nabla E + \lambda \mathbf{w}$$

- El gradiente modificado:

$$\nabla \tilde{E} = \nabla E + \lambda \mathbf{w}$$

- entrenando por descenso de gradiente, en el límite de tiempo continuo (ignorando E):

$$\frac{\partial \mathbf{w}}{\partial \tau} = -\eta \lambda \mathbf{w}$$

- El gradiente modificado:

$$\nabla \tilde{E} = \nabla E + \lambda \mathbf{w}$$

- entrenando por descenso de gradiente, en el límite de tiempo continuo (ignorando E):

$$\frac{\partial \mathbf{w}}{\partial \tau} = -\eta \lambda \mathbf{w}$$

- Solucionando:

$$\mathbf{w}(\tau) \approx \mathbf{w}(0) \exp(-\mu \lambda \tau)$$

- El gradiente modificado:

$$\nabla \tilde{E} = \nabla E + \lambda \mathbf{w}$$

- entrenando por descenso de gradiente, en el límite de tiempo continuo (ignorando E):

$$\frac{\partial \mathbf{w}}{\partial \tau} = -\eta \lambda \mathbf{w}$$

- Solucionando:

$$\mathbf{w}(\tau) \approx \mathbf{w}(0) \exp(-\mu \lambda \tau)$$

- Es decir, los pesos decaen exponencialmente hacia cero.

- El gradiente modificado:

$$\nabla \tilde{E} = \nabla E + \lambda \mathbf{w}$$

- entrenando por descenso de gradiente, en el límite de tiempo continuo (ignorando E):

$$\frac{\partial \mathbf{w}}{\partial \tau} = -\eta \lambda \mathbf{w}$$

- Solucionando:

$$\mathbf{w}(\tau) \approx \mathbf{w}(0) \exp(-\mu \lambda \tau)$$

- Es decir, los pesos decaen exponencialmente hacia cero.
- El algoritmo de aprendizaje para la función de error modificada es una simple extensión de backpropagation.

- El gradiente modificado:

$$\nabla \tilde{E} = \nabla E + \lambda \mathbf{w}$$

- entrenando por descenso de gradiente, en el límite de tiempo continuo (ignorando E):

$$\frac{\partial \mathbf{w}}{\partial \tau} = -\eta \lambda \mathbf{w}$$

- Solucionando:

$$\mathbf{w}(\tau) \approx \mathbf{w}(0) \exp(-\mu \lambda \tau)$$

- Es decir, los pesos decaen exponencialmente hacia cero.
- El algoritmo de aprendizaje para la función de error modificada es una simple extensión de backpropagation.
- Restar $\mu \lambda w_i$ cada vez que el peso w_i es modificado.

Análisis en el caso cuadrático

- Para la función de error cuadrática:

$$E(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{H}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

  El mínimo $\hat{\mathbf{w}}$ satisface:

Análisis en el caso cuadrático

- Para la función de error cuadrática:

$$E(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{H}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

El mínimo $\hat{\mathbf{w}}$ satisface:

$$\mathbf{H}\hat{\mathbf{w}} + \mathbf{b} = 0$$

Análisis en el caso cuadrático

- Para la función de error cuadrática:

$$E(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{H}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

  El mínimo $\hat{\mathbf{w}}$ satisface:

$$\mathbf{H}\hat{\mathbf{w}} + \mathbf{b} = 0$$

- Con weight decay:

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T(\mathbf{H} + \lambda\mathbf{I})\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

Análisis en el caso cuadrático

- Para la función de error cuadrática:

$$E(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{H}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

El mínimo $\hat{\mathbf{w}}$ satisface:

$$\mathbf{H}\hat{\mathbf{w}} + \mathbf{b} = 0$$

- Con weight decay:

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T(\mathbf{H} + \lambda\mathbf{I})\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

El mínimo $\tilde{\mathbf{w}}$ satisface:

$$(\mathbf{H} + \lambda\mathbf{I})\tilde{\mathbf{w}} + b = 0$$

Análisis en el caso cuadrático

- Para la función de error cuadrática:

$$E(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{H}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

El mínimo $\hat{\mathbf{w}}$ satisface:

$$\mathbf{H}\hat{\mathbf{w}} + \mathbf{b} = 0$$

- Con weight decay:

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T(\mathbf{H} + \lambda\mathbf{I})\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

El mínimo $\tilde{\mathbf{w}}$ satisface:

$$(\mathbf{H} + \lambda\mathbf{I})\tilde{\mathbf{w}} + b = 0$$

- Si $\mathbf{H} > 0$, podemos expresar $\hat{\mathbf{w}}$ y $\tilde{\mathbf{w}}$ en términos de los vectores propios ortonormales $\mathbf{u}_i$ (con valores propios α_j) de $\mathbf{H}$:

$$\hat{\mathbf{w}} = \sum_j \hat{w}_j \mathbf{u}_j \qquad \tilde{\mathbf{w}} = \sum_j \tilde{w}_j \mathbf{u}_j$$

- Si $\mathbf{H} > 0$, podemos expresar $\hat{\mathbf{w}}$ y $\tilde{\mathbf{w}}$ en términos de los vectores propios ortonormales $\mathbf{u}_i$ (con valores propios α_j) de $\mathbf{H}$:

$$\hat{\mathbf{w}} = \sum_j \hat{w}_j \mathbf{u}_j \qquad \tilde{\mathbf{w}} = \sum_j \tilde{w}_j \mathbf{u}_j$$

- Reemplazando:

$$\mathbf{b} + \sum_j \hat{w}_j \alpha_j \mathbf{u}_j = 0 \qquad \mathbf{b} + \sum_j \tilde{w}_j \alpha_j \mathbf{u}_j + \lambda \sum \tilde{w}_j \mathbf{u}_j = 0$$

- Si $\mathbf{H} > 0$, podemos expresar $\hat{\mathbf{w}}$ y $\tilde{\mathbf{w}}$ en términos de los vectores propios ortonormales $\mathbf{u}_i$ (con valores propios α_j) de $\mathbf{H}$:

$$\hat{\mathbf{w}} = \sum_j \hat{w}_j \mathbf{u}_j \qquad \tilde{\mathbf{w}} = \sum_j \tilde{w}_j \mathbf{u}_j$$

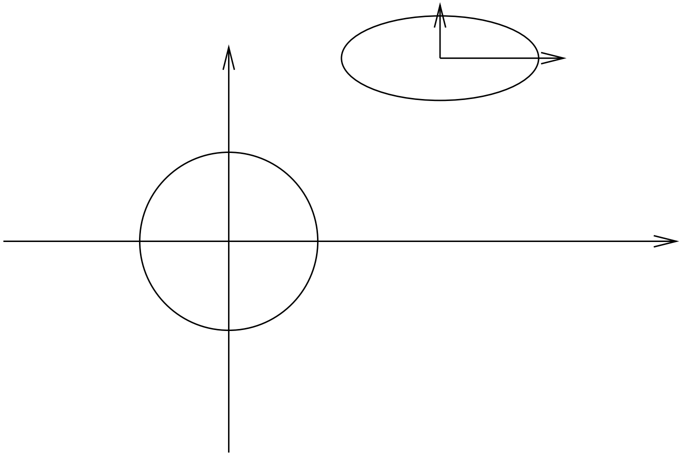
- Reemplazando:

$$\mathbf{b} + \sum_j \hat{w}_j \alpha_j \mathbf{u}_j = 0 \qquad \mathbf{b} + \sum_j \tilde{w}_j \alpha_j \mathbf{u}_j + \lambda \sum \tilde{w}_j \mathbf{u}_j = 0$$

- Por ortonormalidad de los vectores propios:

$$\hat{w}_j \alpha_j = \tilde{w}_j \alpha_j + \lambda \tilde{w}_j \Rightarrow \tilde{w}_j = \hat{w}_j \left(\frac{\alpha_j}{\alpha_j + \lambda} \right)$$

$$\alpha_j \gg \lambda \Rightarrow \tilde{w}_j \approx \hat{w}_j \qquad \alpha_j \ll \lambda \Rightarrow |\tilde{w}_j| \ll |\hat{w}_j|$$



Eliminación de pesos

Eliminación de pesos

- Primo cercano de weight decay.

Eliminación de pesos

- Primo cercano de weight decay.
- La función de error en este caso es:

$$E(\tilde{\mathbf{w}}) = E(\mathbf{w}) + \lambda \sum_i \left(\frac{w_i^2/c^2}{1 + w_i^2/c^2} \right)$$

Eliminación de pesos

- Primo cercano de weight decay.
- La función de error en este caso es:

$$E(\tilde{\mathbf{w}}) = E(\mathbf{w}) + \lambda \sum_i \left(\frac{w_i^2/c^2}{1 + w_i^2/c^2} \right)$$

- Cuando $w_i \gg c$, se tiene $\frac{w_i^2/c^2}{1+w_i^2/c^2} \approx 1$, y el término regularizador cuenta el número de pesos.

Eliminación de pesos

- Primo cercano de weight decay.
- La función de error en este caso es:

$$E(\tilde{\mathbf{w}}) = E(\mathbf{w}) + \lambda \sum_i \left(\frac{w_i^2/c^2}{1 + w_i^2/c^2} \right)$$

- Cuando $w_i \gg c$, se tiene $\frac{w_i^2/c^2}{1+w_i^2/c^2} \approx 1$, y el término regularizador cuenta el número de pesos.
- Cuando $w_i \ll c$, se tiene $\frac{w_i^2/c^2}{1+w_i^2/c^2} \propto w_i^2$ y tenemos weight decay.

Eliminación de pesos

- Primo cercano de weight decay.
- La función de error en este caso es:

$$E(\tilde{\mathbf{w}}) = E(\mathbf{w}) + \lambda \sum_i \left(\frac{w_i^2/c^2}{1 + w_i^2/c^2} \right)$$

- Cuando $w_i \gg c$, se tiene $\frac{w_i^2/c^2}{1+w_i^2/c^2} \approx 1$, y el término regularizador cuenta el número de pesos.
- Cuando $w_i \ll c$, se tiene $\frac{w_i^2/c^2}{1+w_i^2/c^2} \propto w_i^2$ y tenemos weight decay.
- Seleccionando c apropiadamente, podemos forzar a la red a buscar soluciones con unos pocos pesos grandes, o muchos pesos pequeños.

Técnicas de prunning

Técnicas de prunning

- Se trata de encontrar el tamaño apropiado de la red.

Técnicas de prunning

- Se trata de encontrar el tamaño apropiado de la red.
- En prunning se comienza con una red suficientemente grande.

Técnicas de prunning

- Se trata de encontrar el tamaño apropiado de la red.
- En prunning se comienza con una red suficientemente grande.
- Se entrena esta red.

Técnicas de prunning

- Se trata de encontrar el tamaño apropiado de la red.
- En prunning se comienza con una red suficientemente grande.
- Se entrena esta red.
- Se remueven pesos y/o nodos.

Técnicas de prunning

- Se trata de encontrar el tamaño apropiado de la red.
- En prunning se comienza con una red suficientemente grande.
- Se entrena esta red.
- Se remueven pesos y/o nodos.
- Qué es suficientemente grande?

Técnicas de prunning

- Se trata de encontrar el tamaño apropiado de la red.
- En prunning se comienza con una red suficientemente grande.
- Se entrena esta red.
- Se remueven pesos y/o nodos.
- Qué es suficientemente grande?
  - Típicamente dos o tres capas.

Técnicas de prunning

- Se trata de encontrar el tamaño apropiado de la red.
- En prunning se comienza con una red suficientemente grande.
- Se entrena esta red.
- Se remueven pesos y/o nodos.
- Qué es suficientemente grande?
  - Típicamente dos o tres capas.
  - Típicamente un numero de neuronas mucho menor al número de datos de entrenamiento.

Técnicas de prunning

- Se trata de encontrar el tamaño apropiado de la red.
- En prunning se comienza con una red suficientemente grande.
- Se entrena esta red.
- Se remueven pesos y/o nodos.
- Qué es suficientemente grande?
  - Típicamente dos o tres capas.
  - Típicamente un numero de neuronas mucho menor al número de datos de entrenamiento.
  - Número de neuronas polinomial en la dimensión de la entrada.

Estrategia

- Eliminar pesos, pero manteniendo capacidad funcional para resolver el problema.

Estrategia

- Eliminar pesos, pero manteniendo capacidad funcional para resolver el problema.
  - Mejor generalización.

Estrategia

- Eliminar pesos, pero manteniendo capacidad funcional para resolver el problema.
  - Mejor generalización.
  - Entrenamiento más fácil.

Estrategia

- Eliminar pesos, pero manteniendo capacidad funcional para resolver el problema.
  - Mejor generalización.
  - Entrenamiento más fácil.
- Aproximación más simple:

Estrategia

- Eliminar pesos, pero manteniendo capacidad funcional para resolver el problema.
  - Mejor generalización.
  - Entrenamiento más fácil.
- Aproximación más simple:
  1. Entrenar.

Estrategia

- Eliminar pesos, pero manteniendo capacidad funcional para resolver el problema.
  - Mejor generalización.
  - Entrenamiento más fácil.
- Aproximación más simple:
  1. Entrenar.
  2. Eliminar pesos pequeños (comparar con umbral).

Estrategia

- Eliminar pesos, pero manteniendo capacidad funcional para resolver el problema.
  - Mejor generalización.
  - Entrenamiento más fácil.
- Aproximación más simple:
  1. Entrenar.
  2. Eliminar pesos pequeños (comparar con umbral).
- Problema : pesos pequeños pueden tener efecto grande en la función de error.

Saliency

- Medida de cuánto afecta el remover un peso dado a la función de error.

Saliency

- Medida de cuánto afecta el remover un peso dado a la función de error.
- Cerca al mínimo:
$$E \approx \delta\mathbf{w}^T\mathbf{H}\delta\mathbf{w}$$
donde $(\mathbf{H})_{ij} = \frac{\partial^2 E}{\partial w_i w_j}$.

Saliency

- Medida de cuánto afecta el remover un peso dado a la función de error.
- Cerca al mínimo:
$$E \approx \delta \mathbf{w}^T \mathbf{H} \delta \mathbf{w}$$
donde $(\mathbf{H})_{ij} = \frac{\partial^2 E}{\partial w_i w_j}$.
- Asumiendo $\mathbf{H}$ diagonal:

Saliency

- Medida de cuánto afecta el remover un peso dado a la función de error.
- Cerca al mínimo:
$$E \approx \delta \mathbf{w}^T \mathbf{H} \delta \mathbf{w}$$
donde $(\mathbf{H})_{ij} = \frac{\partial^2 E}{\partial w_i w_j}$.
- Asumiendo $\mathbf{H}$ diagonal:
$$E \approx \sum_i \delta w_i^2 (\mathbf{H})_{ii}$$

Saliency

- Medida de cuánto afecta el remover un peso dado a la función de error.
- Cerca al mínimo:

$$E \approx \delta \mathbf{w}^T \mathbf{H} \delta \mathbf{w}$$

  donde $(\mathbf{H})_{ij} = \frac{\partial^2 E}{\partial w_i w_j}$.
- Asumiendo $\mathbf{H}$ diagonal:

$$E \approx \sum_i \delta w_i^2 (\mathbf{H})_{ii}$$

- Es decir

$$w_i \to 0 \quad \Rightarrow \quad \delta E = (\mathbf{H})_{ii} w_i^2$$

Optimal Brain Damage

1. Entrenar red relativamente grande.

Optimal Brain Damage

1. Entrenar red relativamente grande.
2. Entrenar hasta cierto criterio.

Optimal Brain Damage

1. Entrenar red relativamente grande.
2. Entrenar hasta cierto criterio.
3. Calcular $(\mathbf{H})_{ii}$.

Optimal Brain Damage

1. Entrenar red relativamente grande.
2. Entrenar hasta cierto criterio.
3. Calcular $(\mathbf{H})_{ii}$.
4. Ordenar los pesos de acuerdo a $(\mathbf{H})_{ii} w_i^2$

Optimal Brain Damage

1. Entrenar red relativamente grande.
2. Entrenar hasta cierto criterio.
3. Calcular $(\mathbf{H})_{ii}$.
4. Ordenar los pesos de acuerdo a $(\mathbf{H})_{ii}w_i^2$
5. Eliminar pesos con $(\mathbf{H})_{ii}w_i^2$ pequeño.

Optimal Brain Damage

1. Entrenar red relativamente grande.
2. Entrenar hasta cierto criterio.
3. Calcular $(\mathbf{H})_{ii}$.
4. Ordenar los pesos de acuerdo a $(\mathbf{H})_{ii} w_i^2$
5. Eliminar pesos con $(\mathbf{H})_{ii} w_i^2$ pequeño.
6. Volver a entrenar.

Optimal Brain Surgeon

- No asume $\mathbf{H}$ diagonal.

Optimal Brain Surgeon

- No asume $\mathbf{H}$ diagonal.
- Cómo afecta borrar w_i a los otros pesos?

Optimal Brain Surgeon

- No asume $\mathbf{H}$ diagonal.
- Cómo afecta borrar w_i a los otros pesos?
- Hallar $\delta \mathbf{w}$ que minimiza δE sujeto a $w_i = 0$.

Optimal Brain Surgeon

- No asume $\mathbf{H}$ diagonal.
- Cómo afecta borrar w_i a los otros pesos?
- Hallar $\delta\mathbf{w}$ que minimiza δE sujeto a $w_i = 0$.

$$\begin{array}{rl} \text{mín} & \frac{1}{2}\delta\mathbf{w}^T\mathbf{H}\delta\mathbf{w} \\ \text{sujeto a} & \mathbf{e}_i^T w_i + w_i = 0 \end{array}$$

Optimal Brain Surgeon

- No asume $\mathbf{H}$ diagonal.
- Cómo afecta borrar w_i a los otros pesos?
- Hallar $\delta \mathbf{w}$ que minimiza δE sujeto a $w_i = 0$.

$$\begin{aligned} \text{mín} \quad & \frac{1}{2}\delta\mathbf{w}^T\mathbf{H}\delta\mathbf{w} \\ \text{sujeto a} \quad & \mathbf{e}_i^T w_i + w_i = 0 \end{aligned}$$

- El lagrangiano:

$$L(\delta w, \lambda) =$$

Optimal Brain Surgeon

- No asume **H** diagonal.
- Cómo afecta borrar w_i a los otros pesos?
- Hallar $\delta\mathbf{w}$ que minimiza δE sujeto a $w_i = 0$.

$$\begin{aligned} \text{mín} \quad & \frac{1}{2}\delta\mathbf{w}^T\mathbf{H}\delta\mathbf{w} \\ \text{sujeto a} \quad & \mathbf{e}_i^T w_i + w_i = 0 \end{aligned}$$

- El lagrangiano:

$$L(\delta w, \lambda) = \frac{1}{2}\delta w^T\mathbf{H}\delta w +$$

Optimal Brain Surgeon

- No asume **H** diagonal.
- Cómo afecta borrar w_i a los otros pesos?
- Hallar $\delta \mathbf{w}$ que minimiza δE sujeto a $w_i = 0$.

$$\begin{array}{rl} \text{mín} & \frac{1}{2}\delta \mathbf{w}^T \mathbf{H} \delta \mathbf{w} \\ \text{sujeto a} & \mathbf{e}_i^T w_i + w_i = 0 \end{array}$$

- El lagrangiano:

$$L(\delta w, \lambda) = \frac{1}{2}\delta w^T \mathbf{H} \delta w + \lambda(\mathbf{e}_i^T w_i + w_i)$$

- Derivando tenemos:

$$
\begin{aligned}
\nabla L_w &= \mathbf{H}\delta\mathbf{w} + \lambda\mathbf{e}_i = 0 \\
\nabla L_\lambda &= \mathbf{e}_i^T\delta\mathbf{w} + w_i = 0
\end{aligned}
$$

- Derivando tenemos:

$$\nabla L_w = \mathbf{H}\delta\mathbf{w} + \lambda\mathbf{e}_i = 0$$
$$\nabla L_\lambda = \mathbf{e}_i^T\delta\mathbf{w} + w_i = 0$$

- Obtenemos $\lambda = -\frac{w_i}{\mathbf{e}_i^T\mathbf{H}^{-1}\mathbf{e}_i}$ y $\delta\mathbf{w} = \frac{-w_i\mathbf{H}^{-1}\mathbf{e}_i}{[\mathbf{H}^{-1}]_{ii}}$

- Derivando tenemos:

$$\nabla L_w = \mathbf{H}\delta\mathbf{w} + \lambda\mathbf{e}_i = 0$$
$$\nabla L_\lambda = \mathbf{e}_i^T\delta\mathbf{w} + w_i = 0$$

- Obtenemos $\lambda = -\frac{w_i}{\mathbf{e}_i^T\mathbf{H}^{-1}\mathbf{e}_i}$ y $\delta\mathbf{w} = \frac{-w_i\mathbf{H}^{-1}\mathbf{e}_i}{[\mathbf{H}^{-1}]_{ii}}$
- Reemplazando en δE obtenemos el saliency:

$$L_i = \frac{w_i^2}{2(\mathbf{H}^{-1})_{ii}}$$

- Cerca al mínimo $\mathbf{H} \approx \mathbf{J}^T\mathbf{J} \Rightarrow$ se puede aproximar $\mathbf{H}^{-1}$ invirtiendo $\mathbf{J} + \epsilon\mathbf{I}$ para ϵ pequeño, usando fórmula de inversión de Sherman-Morrison-Woodbury.

Optimal Brain Surgeon

1. Entrenar red razonablemente grande hasta error mínimo.

Optimal Brain Surgeon

1. Entrenar red razonablemente grande hasta error mínimo.
2. Calcular $\mathbf{H}^{-1}$.

Optimal Brain Surgeon

1. Entrenar red razonablemente grande hasta error mínimo.
2. Calcular $\mathbf{H}^{-1}$.
3. Encontrar el peso w_i que de el saliency más pequeño.

Optimal Brain Surgeon

1. Entrenar red razonablemente grande hasta error mínimo.
2. Calcular $\mathbf{H}^{-1}$.
3. Encontrar el peso w_i que de el saliency más pequeño.
4. Actualizar todos los pesos con el $\delta\mathbf{w}$ correspondiente.

Optimal Brain Surgeon

1. Entrenar red razonablemente grande hasta error mínimo.
2. Calcular $\mathbf{H}^{-1}$.
3. Encontrar el peso w_i que de el saliency más pequeño.
4. Actualizar todos los pesos con el $\delta\mathbf{w}$ correspondiente.
5. Volver al paso 2 (hasta que no se puedan remover más pesos con incremento significativo del error).

Algoritmos Constructivos

- Determinar el tamaño de la red durante el proceso de entrenamiento.

Algoritmos Constructivos

- Determinar el tamaño de la red durante el proceso de entrenamiento.
- Se comienza con una sola neurona.

Algoritmos Constructivos

- Determinar el tamaño de la red durante el proceso de entrenamiento.
- Se comienza con una sola neurona.
- Se añaden neuronas una a la vez, hasta tener tamaño apropiado.

Algoritmos Constructivos

- Determinar el tamaño de la red durante el proceso de entrenamiento.
- Se comienza con una sola neurona.
- Se añaden neuronas una a la vez, hasta tener tamaño apropiado.
- Se entrena una neurona a la vez: eficiencia computacional.

Algoritmos Constructivos

- Determinar el tamaño de la red durante el proceso de entrenamiento.
- Se comienza con una sola neurona.
- Se añaden neuronas una a la vez, hasta tener tamaño apropiado.
- Se entrena una neurona a la vez: eficiencia computacional.
- Se integra nueva neurona a la red.

Algoritmos Constructivos

- Determinar el tamaño de la red durante el proceso de entrenamiento.
- Se comienza con una sola neurona.
- Se añaden neuronas una a la vez, hasta tener tamaño apropiado.
- Se entrena una neurona a la vez: eficiencia computacional.
- Se integra nueva neurona a la red.
- Teóricamente es posible que el problema de aprendizaje sólo se pueda resolver si se permite añadir pesos y neuronas durante el proceso de entrenamiento.

Ejemplos

- Torre

Ejemplos

- Torre
- Pirámide.

Ejemplos

- Torre
- Pirámide.
- Cascada de correlación.

Ejemplos

- Torre
- Pirámide.
- Cascada de correlación.
- Tiling.

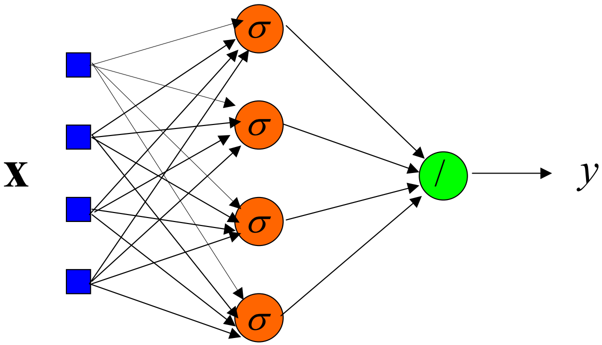
Ejemplos

- Torre
- Pirámide.
- Cascada de correlación.
- Tiling.
- Upstart.

Ejemplos

- Torre
- Pirámide.
- Cascada de correlación.
- Tiling.
- Upstart.

MLP con una capa escondida



$$
\begin{aligned}
y = h(\mathbf{x}) &= f_o\left(a_0 + \sum_{k=1}^{N} a_k f_s(\mathbf{w}_k^T \mathbf{x})\right) \\
&= a_0 + \sum_{k=1}^{N} a_k f_s(\mathbf{w}_k^T \mathbf{x})
\end{aligned}
$$

Algoritmo de Aproximación Iterativa

$$g(\mathbf{x}) = 0$$

Algoritmo de Aproximación Iterativa

$g(\mathbf{x}) = 0$

for $i = 1 : N$ **do**

Algoritmo de Aproximación Iterativa

$g(\mathbf{x}) = 0$

for $i = 1 : N$ **do**

Calcule error residual $e_i = y_i - g(\mathbf{x}_i)$

Algoritmo de Aproximación Iterativa

$g(\mathbf{x}) = 0$
for $i = 1 : N$ **do**
  Calcule error residual $e_i = y_i - g(\mathbf{x}_i)$
  Entrene neurona h para ajustarse a los residuos.

Algoritmo de Aproximación Iterativa

$g(\mathbf{x}) = 0$
for $i = 1 : N$ **do**
    Calcule error residual $e_i = y_i - g(\mathbf{x}_i)$
    Entrene neurona h para ajustarse a los residuos.
    Agregue neurona a la red: $g(\mathbf{x}) = \alpha g(\mathbf{x}) + \beta h(\mathbf{x})$, donde α, β minimizan:

$$\sum_{j=1}^{n} (g(\mathbf{x}_j) - y_j)^2$$

Algoritmo de Aproximación Iterativa

$g(\mathbf{x}) = 0$

for $i = 1 : N$ **do**

Calcule error residual $e_i = y_i - g(\mathbf{x}_i)$

Entrene neurona h para ajustarse a los residuos.

Agregue neurona a la red: $g(\mathbf{x}) = \alpha g(\mathbf{x}) + \beta h(\mathbf{x})$, donde α, β minimizan:

$$\sum_{j=1}^{n} (g(\mathbf{x}_j) - y_j)^2$$

end for

Algoritmo de reajuste

for $i = 1 : N$ **do**

Algoritmo de reajuste

for $i = 1 : N$ **do**

  Calcule $\bar{g}(\mathbf{x}) = \sum_{j \neq i}^{N} a_j h_j(\mathbf{x})$

Algoritmo de reajuste

for $i = 1 : N$ **do**

Calcule $\bar{g}(\mathbf{x}) = \sum_{j \neq i}^{N} a_j h_j(\mathbf{x})$

Calcule error residual: $e_i = y_i - \bar{g}(\mathbf{x}_i)$

Algoritmo de reajuste

for $i = 1 : N$ **do**

Calcule $\bar{g}(\mathbf{x}) = \sum_{j \neq i}^{N} a_j h_j(\mathbf{x})$

Calcule error residual: $e_i = y_i - \bar{g}(\mathbf{x}_i)$

Entrene neurona h para ajustarse a los residuos.

Algoritmo de reajuste

for $i = 1 : N$ **do**

Calcule $\bar{g}(\mathbf{x}) = \sum_{j \neq i}^{N} a_j h_j(\mathbf{x})$

Calcule error residual: $e_i = y_i - \bar{g}(\mathbf{x}_i)$

Entrene neurona h para ajustarse a los residuos.

Agregue neurona a la red: $g(\mathbf{x}) = \sum_{i=1}^{n} a_j h_j(\mathbf{x})$, donde $a_1, a_2, \ldots, a_n$ minimizan:

$$\sum_{j=1}^{n} (g(\mathbf{x}_j) - y_j)^2$$

Algoritmo de reajuste

for $i = 1 : N$ **do**

Calcule $\bar{g}(\mathbf{x}) = \sum_{j \neq i}^{N} a_j h_j(\mathbf{x})$

Calcule error residual: $e_i = y_i - \bar{g}(\mathbf{x}_i)$

Entrene neurona h para ajustarse a los residuos.

Agregue neurona a la red: $g(\mathbf{x}) = \sum_{i=1}^{n} a_j h_j(\mathbf{x})$, donde $a_1, a_2, \ldots, a_n$ minimizan:

$$\sum_{j=1}^{n} (g(\mathbf{x}_j) - y_j)^2$$

end for