

Desarrollo de modelos de Deep Learning basados en redes neuronales recurrentes para clasificación de electrocardiogramas

Juan José Herrera Aranda

September 7, 2022



- 1 Motivation
- 2 Aprendizaje Estadístico
- 3 Teorema de Aproximación Universal
- 4 Redes Neuronales
- 5 Descripción del problema
- 6 Resultados y Análisis
- 7 Conclusiones

Outline

1 Motivation

2 Aprendizaje Estadístico

3 Teorema de Aproximación Universal

4 Redes Neuronales

5 Descripción del problema

6 Resultados y Análisis

7 Conclusiones

Motivation ○ ● ○ ○ ○

Arritmia

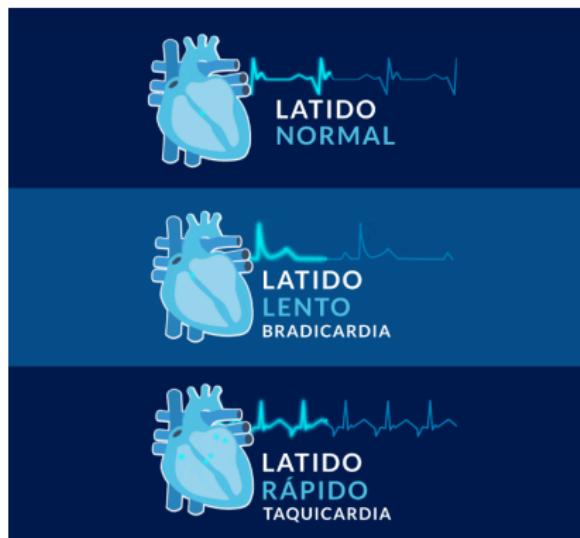
¿Qué es la arritmia?

Arritmia

¿Qué es la arritmia?

Trastorno de la frecuencia (ritmo) cardíaca.

- Taquicardia
- Bradicardia
- **Fibrilación Auricular**



Fuente

Motivation ○ ○ ● ○ ○

Arritmia

¿Es peligrosa?

Arritmia

¿Es peligrosa?

Por lo general...

- Depende del tipo de arritmia.
- Hay algunas que no.
- Pero si es una fibrilación auricular, entonces sí.



Fuente

Motivation ○ ○ ○ ● ○

Arritmia

¿Cómo de peligrosa es la fibrilación auricular?

Arritmia

¿Cómo de peligrosa es la fibrilación auricular?

- Afecta al 1 – 2% de la población.
- No entiende de géneros.
- Su incidencia aumenta con la edad.
- En los próximos 30 años su incidencia aumentará el triple.
- Asociada a una alta tasa de mortandad y morbilidad.

Arritmia

¿Cómo de peligrosa es la fibrilación auricular?

- Afecta al 1 – 2% de la población.
- No entiende de géneros.
- Su incidencia aumenta con la edad.
- En los próximos 30 años su incidencia aumentará el triple.
- Asociada a una alta tasa de mortandad y morbilidad.

Puede causar

- Cansancio y fatiga
- Accidente cerebrovascular.
- Infarto cerebral
- Y en el peor de los casos...

Arritmia

¿Cómo de peligrosa es la fibrilación auricular?

- Afecta al 1 – 2% de la población.
- No entiende de géneros.
- Su incidencia aumenta con la edad.
- En los próximos 30 años su incidencia aumentará el triple.
- Asociada a una alta tasa de mortandad y morbilidad.

Puede causar

- Cansancio y fatiga
- Accidente cerebrovascular.
- Infarto cerebral
- Y en el peor de los casos...



Fuente

Motivation ○ ○ ○ ○ ●

Arritmia

¿Cómo la detectamos?

¿Cómo la detectamos?

- De eso trata este proyecto.

¿Cómo la detectamos?

- De eso trata este proyecto.
- Primeramente, vayamos a los fundamentos de todo...

¿Cómo la detectamos?

- De eso trata este proyecto.
- Primeramente, vayamos a los fundamentos de todo... *las matemáticas*

¿Cómo la detectamos?

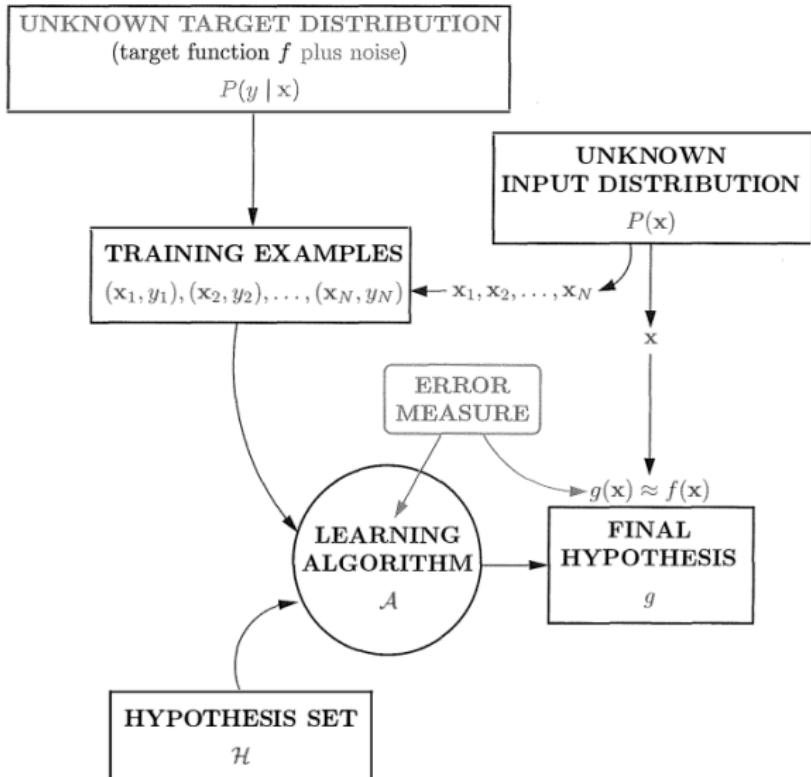
- De eso trata este proyecto.
- Primeramente, vayamos a los fundamentos de todo... *las matemáticas*
- Después, a las técnicas de aprendizaje profundo.

¿Cómo la detectamos?

- De eso trata este proyecto.
- Primeramente, vayamos a los fundamentos de todo... *las matemáticas*
- Después, a las técnicas de aprendizaje profundo.
- Y por último, nos enfrentamos a las fibrilaciones auriculares.

- 1 Motivación
 - 2 Aprendizaje Estadístico
 - 3 Teorema de Aproximación Universal
 - 4 Redes Neuronales
 - 5 Descripción del problema
 - 6 Resultados y Análisis
 - 7 Conclusiones

Introducción



Definition (*Error Empírico*)

Dada $h \in \mathcal{H}$ y $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ un conjunto de entrenamiento, se define el error empírico como

$$\hat{R}_S(h) := \frac{1}{m} \sum_{i=1}^m [\mathbb{I}_{h(x_i) \neq y_i}] \quad (1)$$

Definition (*Error Empírico*)

Dada $h \in \mathcal{H}$ y $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ un conjunto de entrenamiento, se define el error empírico como

$$\hat{R}_S(h) := \frac{1}{m} \sum_{i=1}^m [\mathbb{I}_{h(x_i) \neq y_i}] \quad (1)$$

Definition (*Error de Generalización - Clasificación Binaria*)

Dada $h \in \mathcal{H}$, una función objetivo f y una distribución de probabilidad P definimos el error de generalización de h como

$$R_{P,f}(h) := \mathbb{P}_{x \sim P}[h(x) \neq f(x)] = P(\{x : h(x) \neq f(x)\}) \quad \forall x \in \mathcal{X} \quad (2)$$

Definition (*Error Empírico*)

Dada $h \in \mathcal{H}$ y $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ un conjunto de entrenamiento, se define el error empírico como

$$\hat{R}_S(h) := \frac{1}{m} \sum_{i=1}^m [\mathbb{I}_{h(x_i) \neq y_i}] \quad (1)$$

Definition (*Error de Generalización - Clasificación Binaria*)

Dada $h \in \mathcal{H}$, una función objetivo f y una distribución de probabilidad P definimos el error de generalización de h como

$$R_{P,f}(h) := \mathbb{P}_{x \sim P}[h(x) \neq f(x)] = P(\{x : h(x) \neq f(x)\}) \quad \forall x \in \mathcal{X} \quad (2)$$

Criterio ERM

Se elige una hipótesis $h_S \in \mathcal{H}$ tal que $h_S \in \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}_S(h)$

Postulate (*Asunción de realizabilidad*)

Se supone que existe $h^* \in \mathcal{H}$ tal que $R_{P,f}(h^*) = 0$

Aprendizaje Correcto Probablemente Aproximado

Postulate (*Asunción de realizabilidad*)

Se supone que existe $h^* \in \mathcal{H}$ tal que $R_{P,f}(h^*) = 0$

Definition (*Aprendizaje PAC*)

Una clase de hipótesis \mathcal{H} es PAC aprendible cuando exista una función $m_{\mathcal{H}} : (0, 1)^2 \rightarrow \mathbb{N}$ y un algoritmo de aprendizaje \mathcal{A} con la siguiente propiedad: Para todo $\epsilon, \delta \in (0, 1)$, para cada distribución P sobre \mathcal{X} y para cada función objetivo $f : \mathcal{X} \rightarrow \{0, 1\}$, si se cumple la hipótesis de realizabilidad con respecto \mathcal{H}, P, f , entonces cuando ejecutamos el algoritmo de aprendizaje \mathcal{A} en $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. ejemplos generados por P y etiquetados por f , \mathcal{A} devuelve una hipótesis $h \in \mathcal{H}$ tal que,

$$\mathbb{P}_{S \sim P^m}[R_{P,f}(h) \leq \epsilon] \geq 1 - \delta \quad (3)$$

Proposition

Bajo la asunción de realizabilidad, toda clase de hipótesis finita es PAC aprendible verificando,

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \rceil$$

Proposition

Bajo la asunción de realizabilidad, toda clase de hipótesis finita es PAC aprendible verificando,

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \rceil$$

pero...

Proposition

Bajo la asunción de realizabilidad, toda clase de hipótesis finita es PAC aprendible verificando,

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \rceil$$

¿Qué pasa si eliminamos la asunción de realizabilidad?

Proposition

Bajo la asunción de realizable, toda clase de hipótesis finita es PAC aprendible verificando,

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \rceil$$

¿Qué pasa si eliminamos la asunción de realizable?

¿Qué pasa cuando la clase de hipótesis es infinita?

Proposition

Bajo la asunción de realizable, toda clase de hipótesis finita es PAC aprendible verificando,

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \rceil$$

¿Qué pasa si eliminamos la asunción de realizable?

Aprendizaje PAC Agnóstico + Convergencia Uniforme

¿Qué pasa cuando la clase de hipótesis es infinita?

Dimensión de Vapnik-Chervonenkis

Aprendizaje Estadístico ○ ○ ○ ○ ● ○ ○ ○ ○ ○

Aprendizaje Correcto Probablemente Aproximado

Aprendizaje PAC Agnóstico

Eliminamos la asunción de realizabilidad y definimos...

Aprendizaje PAC Agnóstico

Definition (Aprendizaje PAC Agnóstico)

Una clase de hipótesis \mathcal{H} es PAC Agnóstica con respecto a un conjunto $Z \subset \mathcal{X} \times \mathcal{Y}$ y una función de pérdida medible $\ell : \mathcal{H} \times Z \rightarrow \mathbb{R}^+$ cuando existe una función $m_{\mathcal{H}} : (0, 1)^2 \rightarrow \mathbb{N}$ y un algoritmo de aprendizaje \mathcal{A} cumpliendo la siguiente propiedad: Para cada $\epsilon, \delta \in (0, 1)$ y para cada distribución P sobre Z , cuando ejecutamos el algoritmo de aprendizaje sobre $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ ejemplos *i.i.d.* generados por P , el algoritmo devuelve $h \in \mathcal{H}$ tal que,

$$\mathbb{P}_{S \sim P}[L_P(h) - \min_{h' \in \mathcal{H}} L_P(h') \leq \epsilon] \geq 1 - \delta \quad (4)$$

donde $L_P(h) = \mathbb{E}_{z \sim P}[\ell(h, z)]$

Aprendizaje Correcto Probablemente Aproximado

Convergencia Uniforme

Definition (*Convergencia uniforme*)

Decimos que una clase de hipótesis \mathcal{H} cumple la propiedad de la convergencia uniforme (con respecto a un dominio Z y una función de error ℓ) cuando existe una función $m_{\mathcal{H}}^{UC} : (0, 1)^2 \rightarrow \mathbb{N}$ tal que para cada $\epsilon, \delta \in (0, 1)$ y para cada distribución de probabilidad, P , sobre Z , si el conjunto de entrenamiento, S , tiene $m \geq m_{\mathcal{H}}^{UC}(\epsilon, \delta)$ ejemplos tomados de manera (*i.i.d.*) de acuerdo a P , entonces con probabilidad al menos $1 - \delta$ se cumple,

$$|\hat{R}_S(h) - R_{P,f}(h)| \leq \epsilon \quad \forall h \in \mathcal{H} \quad (5)$$

Resultado

Proposition

Si una clase de hipótesis cumple la propiedad de convergencia uniforme entonces es PAC agnóstica aprendible con una complejidad de la muestra $m_H(\epsilon, \delta) \leq m_{\mathcal{H}}^{UC}\left(\frac{\epsilon}{2}, \delta\right)$.

Dimensión Vapnik-Chervonenkis

Definiciones Principales

Definition (*función de crecimiento*)

Sea \mathcal{H} una clase de hipótesis. La función de crecimiento de \mathcal{H} , $m_{\mathcal{H}} : \mathbb{N} \rightarrow \mathbb{N}$ se define como

$$m_{\mathcal{H}}(N) = \max_{x_1, \dots, x_N \in \mathcal{X}} |\mathcal{H}(x_1, \dots, x_N)| \quad (6)$$

Dimensión Vapnik-Chervonenkis

Definiciones Principales

Definition (*función de crecimiento*)

Sea \mathcal{H} una clase de hipótesis. La función de crecimiento de \mathcal{H} , $m_{\mathcal{H}} : \mathbb{N} \rightarrow \mathbb{N}$ se define como

$$m_{\mathcal{H}}(N) = \max_{x_1, \dots, x_N \in \mathcal{X}} |\mathcal{H}(x_1, \dots, x_N)| \quad (6)$$

Definition (*Dimensión Vapnik-Chervonenkis*)

La dimensión VC de una clase de hipótesis \mathcal{H} se define como $d_{VC}(\mathcal{H}) = \max\{N \in \mathbb{N} : m_{\mathcal{H}}(N) = 2^N\}$. Si $m_{\mathcal{H}}(N) = 2^N$ para todo N , entonces $d_{VC}(\mathcal{H}) = \infty$.

Dimensión Vapnik-Chervonenkis

Definiciones Principales

Definition (*función de crecimiento*)

Sea \mathcal{H} una clase de hipótesis. La función de crecimiento de \mathcal{H} , $m_{\mathcal{H}} : \mathbb{N} \rightarrow \mathbb{N}$ se define como

$$m_{\mathcal{H}}(N) = \max_{x_1, \dots, x_N \in \mathcal{X}} |\mathcal{H}(x_1, \dots, x_N)| \quad (6)$$

Definition (*Dimensión Vapnik-Chervonenkis*)

La dimensión VC de una clase de hipótesis \mathcal{H} se define como $d_{VC}(\mathcal{H}) = \max\{N \in \mathbb{N} : m_{\mathcal{H}}(N) = 2^N\}$. Si $m_{\mathcal{H}}(N) = 2^N$ para todo N , entonces $d_{VC}(\mathcal{H}) = \infty$.

Proposition

Sea \mathcal{H} una clase de hipótesis tal que $d_{VC}(\mathcal{H}) = \infty$, entonces \mathcal{H} no es PAC aprendible.

Teorema Fundamental del Aprendizaje Estadístico

Primer resultado más importante

Theorem (*Teorema Fundamental del aprendizaje estadístico*)

Sea \mathcal{H} una clase de hipótesis que van de \mathcal{X} a $\{0, 1\}$ y sea la función de pérdida $\ell = \ell_{0-1}$. Equivalen las siguientes afirmaciones:

- ① \mathcal{H} cumple la propiedad de convergencia uniforme
- ② El criterio ERM garantiza que \mathcal{H} sea PAC agnóstica aprendible
- ③ \mathcal{H} es PAC agnóstica aprendible
- ④ \mathcal{H} es PAC aprendible
- ⑤ El criterio ERM garantiza que \mathcal{H} sea PAC aprendible
- ⑥ \mathcal{H} tiene una dimensión VC finita

Cotas de Generalización

Partimos de la desigualdad de Hoeffding:

$$\mathbb{P}\left[\left|\hat{R}_S(g) - R_P(g)\right| \geq \epsilon\right] \leq 2e^{-2m\epsilon^2}$$

Cotas de Generalización

Partimos de la desigualdad de Hoeffding:

$$\mathbb{P}\left[\left|\hat{R}_S(g) - R_P(g)\right| \geq \epsilon\right] \leq 2e^{-2m\epsilon^2}$$

- Fijada una hipótesis $g \in \mathcal{H}$:

$$R_P(g) \leq \hat{R}_S(g) + \sqrt{\frac{1}{2m} \log \frac{2}{\delta}}$$

Cotas de Generalización

Partimos de la desigualdad de Hoeffding:

$$\mathbb{P}\left[\left|\hat{R}_S(g) - R_P(g)\right| \geq \epsilon\right] \leq 2e^{-2m\epsilon^2}$$

- Fijada una hipótesis $g \in \mathcal{H}$:

$$R_P(g) \leq \hat{R}_S(g) + \sqrt{\frac{1}{2m} \log \frac{2}{\delta}}$$

- \mathcal{H} finito y consistente:

$$R_P(h_S) \leq \frac{1}{m} \left(\log \left(\frac{|\mathcal{H}|}{\delta} \right) \right)$$

Cotas de Generalización

Partimos de la desigualdad de Hoeffding:

$$\mathbb{P}\left[\left|\hat{R}_S(g) - R_P(g)\right| \geq \epsilon\right] \leq 2e^{-2m\epsilon^2}$$

- Fijada una hipótesis $g \in \mathcal{H}$:

$$R_P(g) \leq \hat{R}_S(g) + \sqrt{\frac{1}{2m} \log \frac{2}{\delta}}$$

- \mathcal{H} finito y consistente:

$$R_P(h_S) \leq \frac{1}{m} \left(\log \left(\frac{|\mathcal{H}|}{\delta} \right) \right)$$

- \mathcal{H} finito y no consistente:

$$\forall h \in \mathcal{H}, \quad R(h) \leq \hat{R}(h) + \sqrt{\frac{1}{2m} \log \left(\frac{2|\mathcal{H}|}{\delta} \right)}$$

Cotas de Generalización

Segundo resultado más importante

Theorem (*Cota de generalización VC*)

Para cualquier tolerancia $\delta \in (0, 1)$, con probabilidad al menos $1 - \delta$ ocurre,

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{8}{N} \log \left(\frac{4m_{\mathcal{H}}(2N)}{\delta} \right)} \quad \forall h \in \mathcal{H} \quad (7)$$

Cotas de Generalización

Segundo resultado más importante

Theorem (*Cota de generalización VC*)

Para cualquier tolerancia $\delta \in (0, 1)$, con probabilidad al menos $1 - \delta$ ocurre,

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{8}{N} \log \left(\frac{4m_{\mathcal{H}}(2N)}{\delta} \right)} \quad \forall h \in \mathcal{H} \quad (7)$$

- Resultado universal.
- La cota no es fina.
- Establece la viabilidad del aprendizaje con \mathcal{H} infinitos.
- Útil para comparar el rendimiento de generalización de distintos modelos.

Outline

- 1 Motivation
- 2 Aprendizaje Estadístico
- 3 Teorema de Aproximación Universal
- 4 Redes Neuronales
- 5 Descripción del problema
- 6 Resultados y Análisis
- 7 Conclusiones

Antecedentes

- Teorema de Hahn-Banach
- Teorema de Representación de Riesz para funcionales lineales en $C_0(X)$

Corollary

Sea X un espacio normado, M un subespacio de X . Si $x_0 \notin \overline{M}$, existe un $f \in X^*$ tal que $f(x) = 0$ para todo $x \in M$, $f(x_0) = 1$ y $\|f\| = \frac{1}{d}$ donde $d = \text{dist}(x_0, M)$

Resultado

Theorem (Teorema de la Aproximación Universal - G.Cybenko)

Sea σ una función continua sigmoidal. Entonces la suma finita dada por

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(x_j^T x + \theta_j) \quad (8)$$

es densa en $C(I_n)$. En otras palabras, dada cualquier función $f \in C(I_n)$ y $\epsilon > 0$, existe una suma $G(X)$ que viene dada por la expresión de arriba de tal manera que

$$|G(x) - f(x)| < \epsilon \quad \forall x \in I_n \quad (9)$$

Resultado

Theorem (Teorema de la Aproximación Universal - G.Cybenko)

Sea σ una función continua sigmoidal. Entonces la suma finita dada por

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(x_j^T x + \theta_j) \quad (10)$$

es densa en $C(I_n)$. En otras palabras, dada cualquier función $f \in C(I_n)$ y $\epsilon > 0$, existe una suma $G(X)$ que viene dada por la expresión de arriba de tal manera que

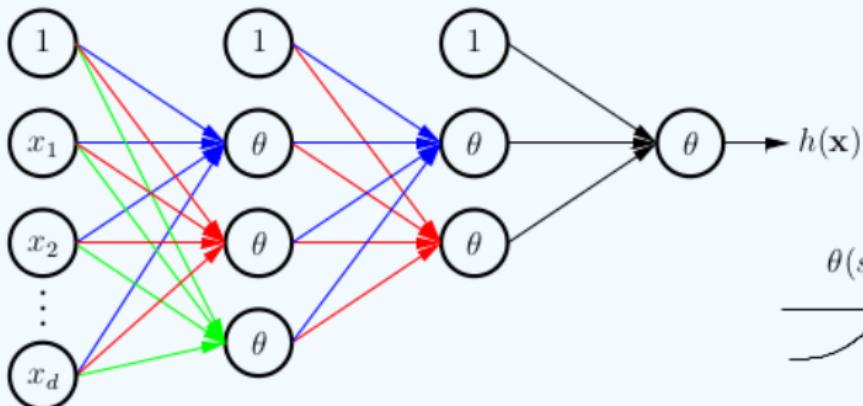
$$|G(x) - f(x)| < \epsilon \quad \forall x \in I_n \quad (11)$$

- Existe una versión para problemas de clasificación
- Existen versiones con otras funciones de activación (ReLU, tanh,...)

Outline

- 1 Motivation
- 2 Aprendizaje Estadístico
- 3 Teorema de Aproximación Universal
- 4 Redes Neuronales
- 5 Descripción del problema
- 6 Resultados y Análisis
- 7 Conclusiones

Redes Neuronales Prealmacenadas

input layer $\ell = 0$ hidden layers $0 < \ell < L$ output layer $\ell = L$

Entrenamiento:

- Optimización → Método del Gradiente Descendiente
- Cálculo del gradiente → Algoritmo BackPropagation

Redes Neuronales Convolucionales

Operación de Convolución - Señales

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} =$$

Redes Neuronales Convolucionales

Operación de Convolución - Señales

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} =$$

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0}$$

Redes Neuronales Convolucionales

Operación de Convolución - Señales

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} =$$

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0}$$

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0} \boxed{30}$$

Redes Neuronales Convolucionales

Operación de Convolución - Señales

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} =$$

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0}$$

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0} \boxed{30}$$

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0} \boxed{30} \boxed{30} \boxed{0}$$

Redes Neuronales Convolucionales

Operación de Convolución - Señales

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} =$$

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0}$$

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0} \boxed{30}$$

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0} \boxed{30} \boxed{30} \boxed{0}$$

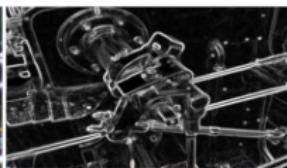
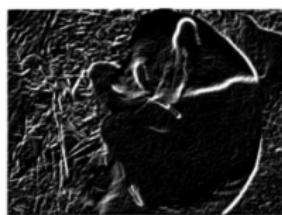
$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \boxed{0} \boxed{30} \boxed{30} \boxed{0}$$

Redes Neuronales Convolucionales

Operación de Convolución - Imágenes

$$\begin{array}{|c|c|c|c|c|c|c|} \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline 0 & 30 & 30 & 0 \\ \hline 0 & 30 & 30 & 0 \\ \hline 0 & 30 & 30 & 0 \\ \hline 0 & 30 & 30 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline \text{White} & \text{Dark} \\ \hline \end{array} * \begin{array}{|c|c|} \hline \text{White} & \text{Dark} \\ \hline \end{array} = \begin{array}{|c|c|} \hline \text{White} & \text{Dark} \\ \hline \end{array}$$

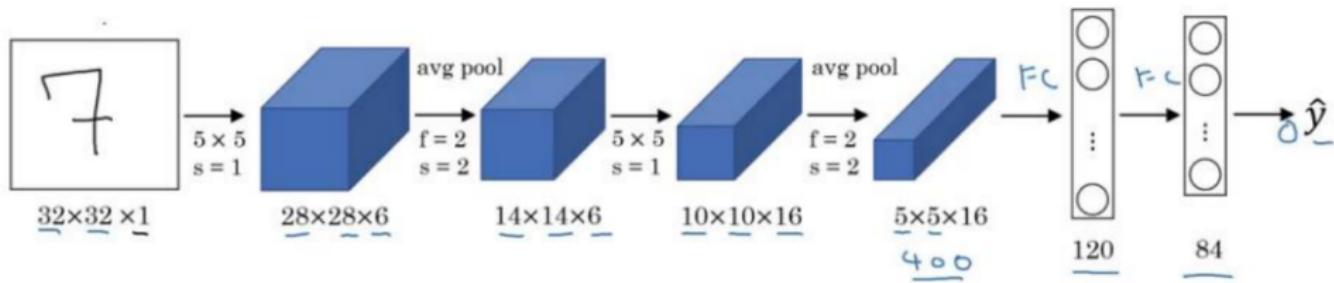


Parámetros de la convolución:

- Zancada (Stride)
- Dilatación
- Tamaño del kernel
- Padding

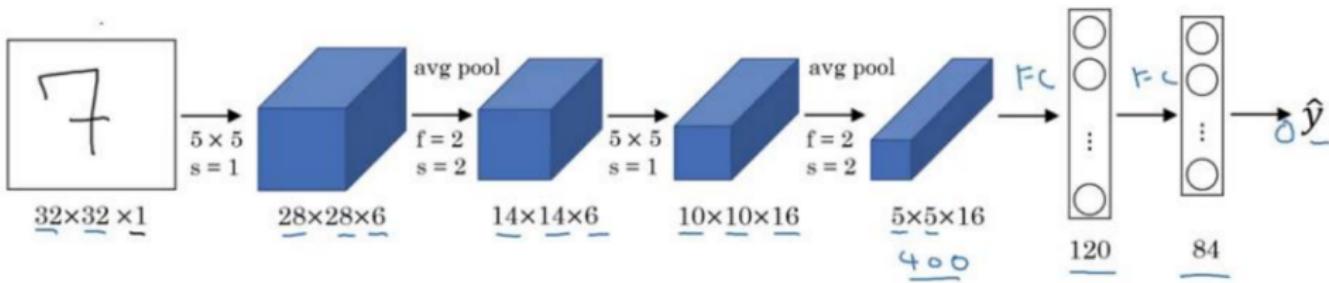
Redes Neuronales Convolucionales

Idea General de una CNN



Redes Neuronales Convolucionales

Idea General de una CNN

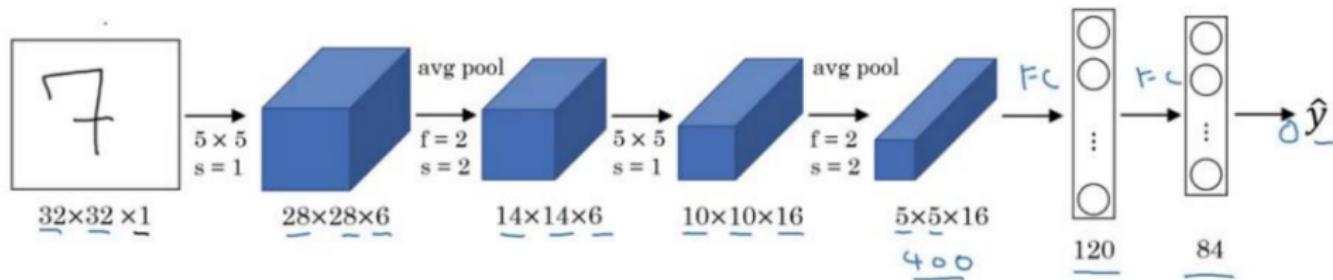


Capas de una CNN

- Capas de convolución
- Capas de pooling
- Capas de normalización
- Capas totalmente conectadas.

Redes Neuronales Convolucionales

Idea General de una CNN



Capas de una CNN

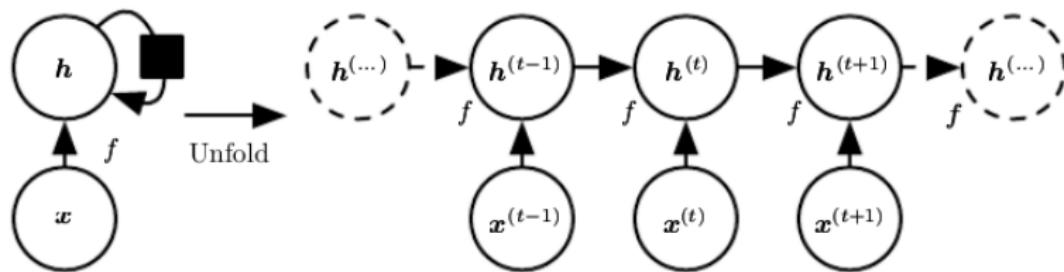
- Capas de convolución
- Capas de pooling
- Capas de normalización
- Capas totalmente conectadas.



Ejemplos de filtros aprendidos por el modelo AlexNet en su primera capa.
(ImageNet2012)

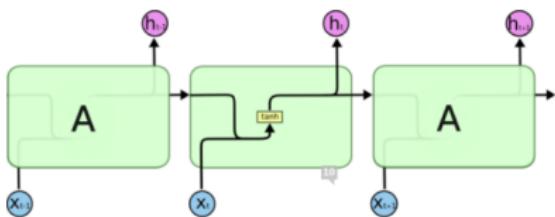
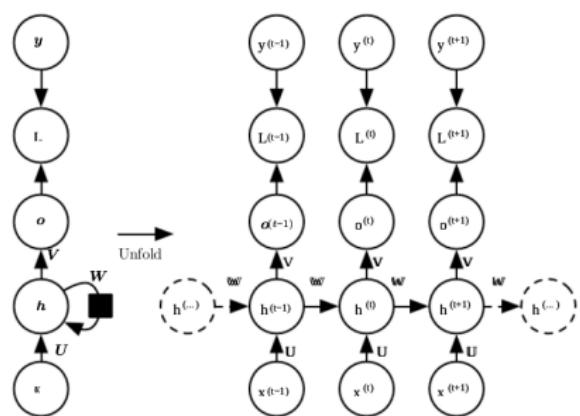
Redes Neuronales Recurrentes

Neurona Recurrente



Redes Neuronales Recurrentes

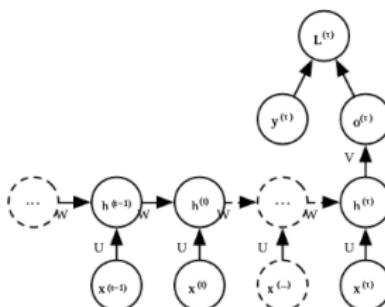
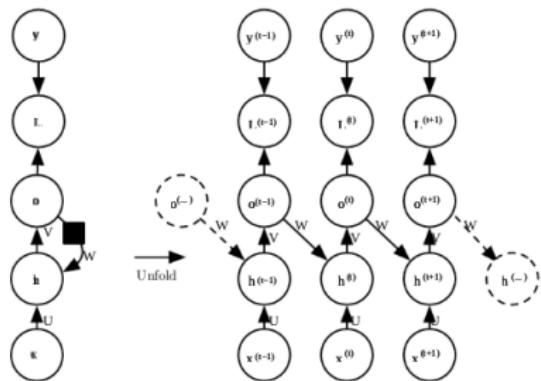
Arquitectura típica de RNN



$$\begin{aligned}
 a^{(t)} &= b + Wh^{(t-1)} + Ux^{(t)}, \\
 h^{(t)} &= \tanh(a^{(t)}), \\
 o^{(t)} &= c + Vh^{(t)}, \\
 \hat{y}^{(t)} &= \text{softmax}(o^{(t)})
 \end{aligned}$$

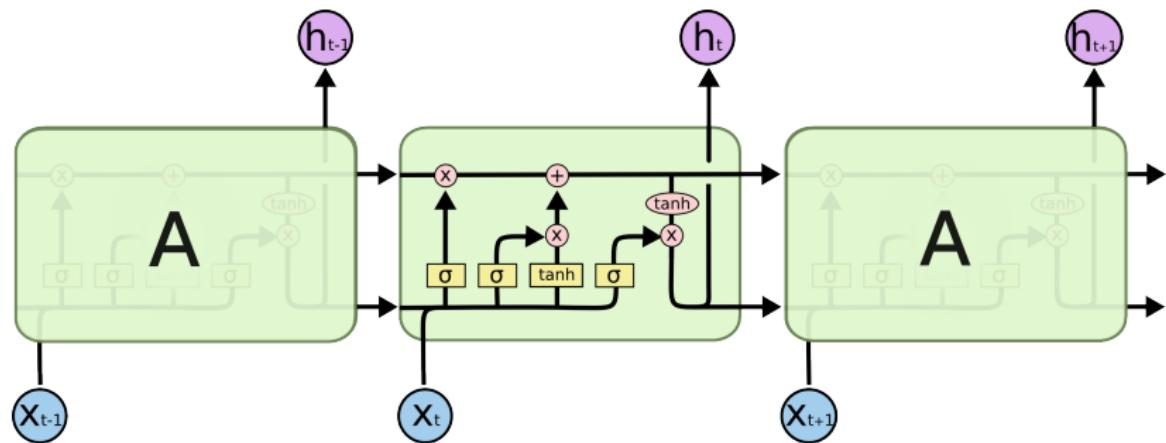
Redes Neuronales Recurrentes

Otras arquitecturas



Redes Neuronales Recurrentes

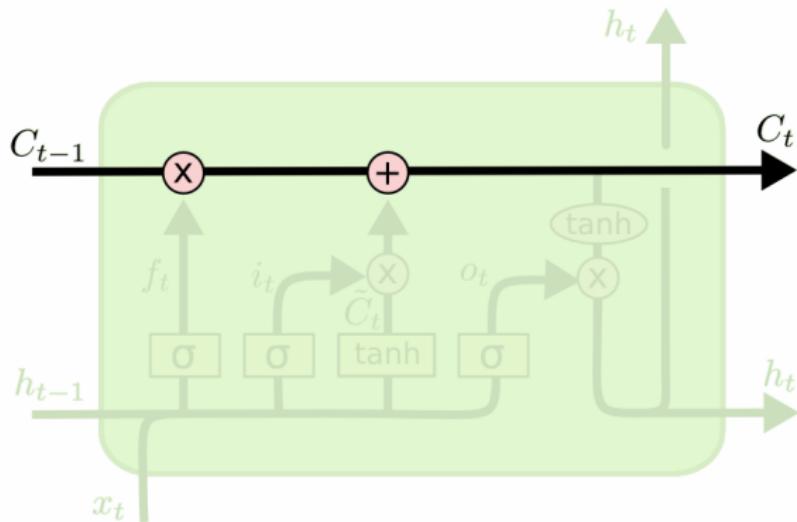
Long Short Term Memory (LSTM)



Redes Neuronales Recurrentes

Long Short Term Memory (LSTM)

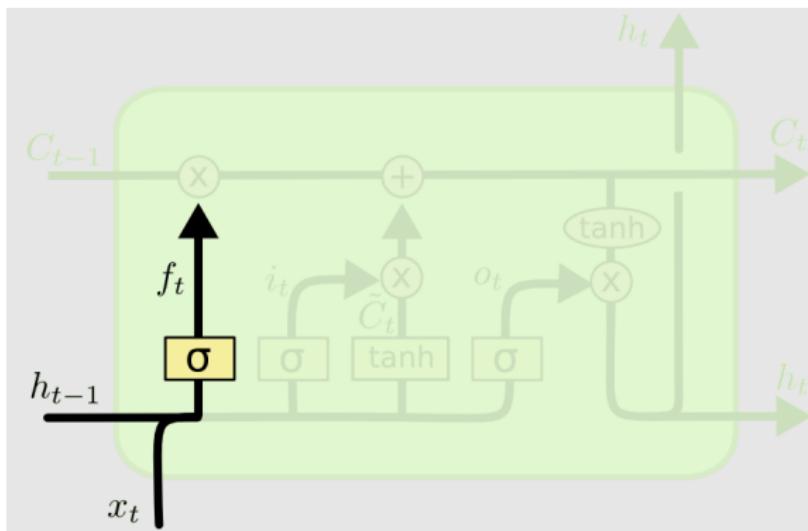
Estado de la celda



Redes Neuronales Recurrentes

Long Short Term Memory (LSTM)

Puerta de Olvido

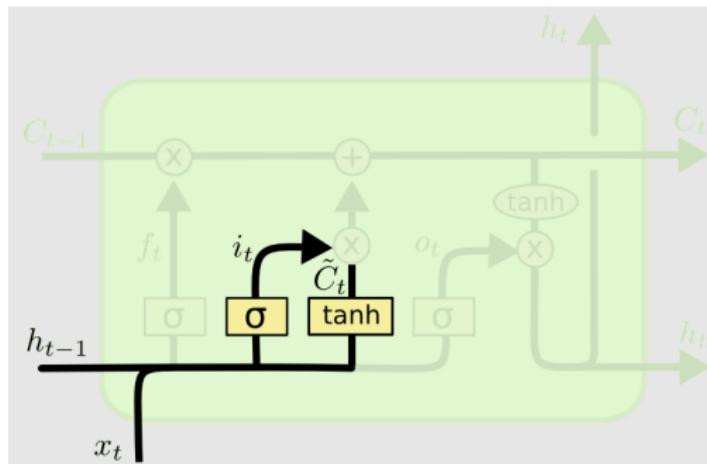


$$f_i^{(t)} = \sigma \left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)} \right)$$

Redes Neuronales Recurrentes

Long Short Term Memory (LSTM)

Puerta de entrada



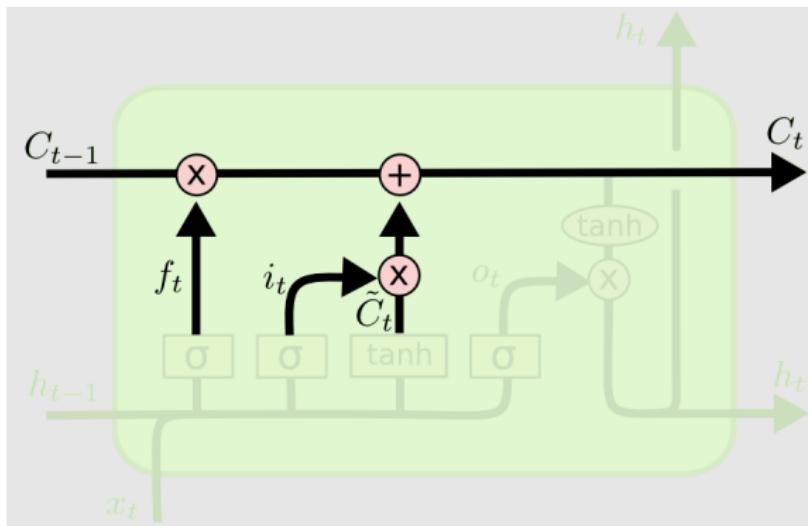
$$g_i^{(t)} = \sigma \left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)} \right)$$

$$\tilde{C}^{(t)} = \tanh \left(b_i^C + \sum_j U_{i,j}^C x_j^{(t)} + \sum_j W_{i,j}^C h_j^{(t-1)} \right)$$

Redes Neuronales Recurrentes

Long Short Term Memory (LSTM)

Actualización

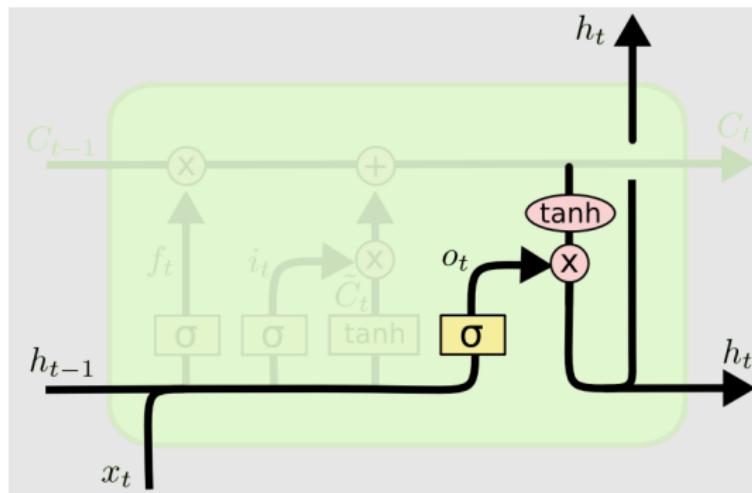


$$C_i^{(t)} = f_i^{(t)} * C_i^{(t-1)} + g_i^{(t)} * \tilde{C}_i^{(t)}$$

Redes Neuronales Recurrentes

Long Short Term Memory (LSTM)

Puerta de salida

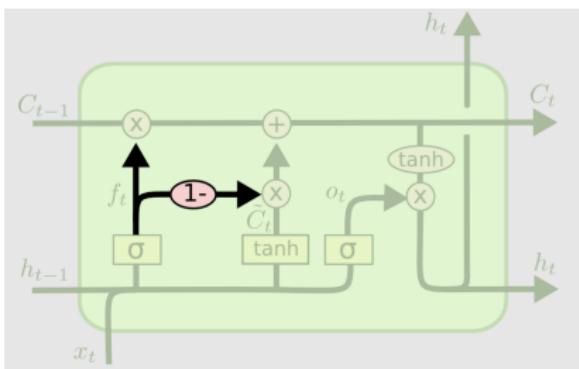
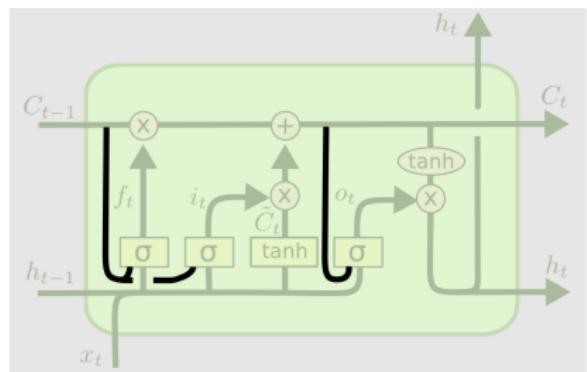


$$o_i^{(t)} = \sigma \left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)} \right)$$

$$h_i^{(t)} = \tanh(C_i^{(t)}) * o_i^{(t)}$$

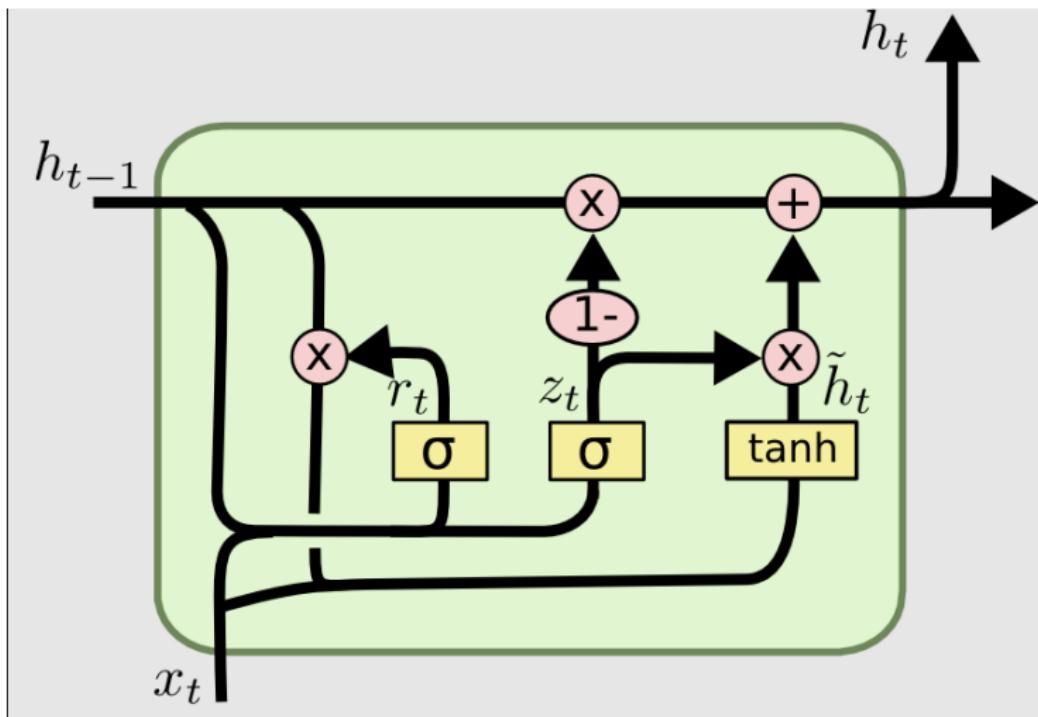
Redes Neuronales Recurrentes

Variaciones LSTM



Redes Neuronales Recurrentes

Gated Recurrent Unit (GRU)



Descripción del problema

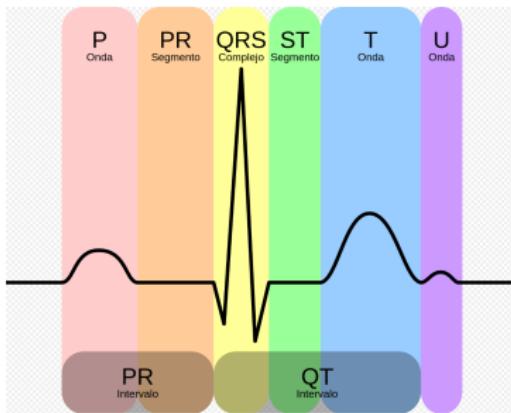
● ○ ○ ○ ○ ○ ○

Outline

- 1 Motivation
- 2 Aprendizaje Estadístico
- 3 Teorema de Aproximación Universal
- 4 Redes Neuronales
- 5 Descripción del problema
- 6 Resultados y Análisis
- 7 Conclusiones

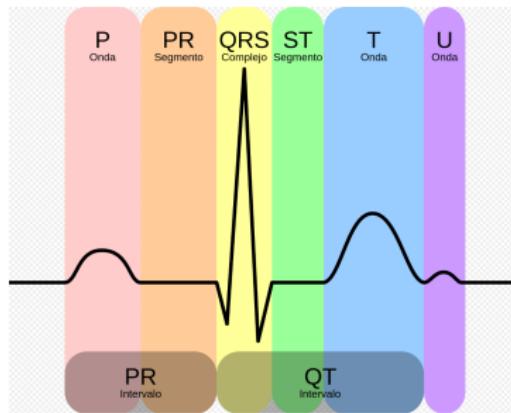
Descripción del problema

Descripción del problema



Descripción del problema

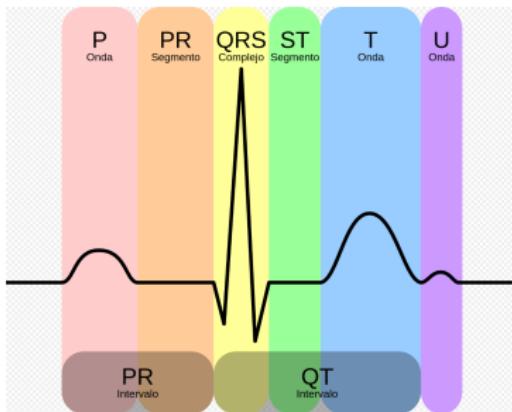
Descripción del problema



- Problema de clasificación de señales obtenidas por ECG.

Descripción del problema

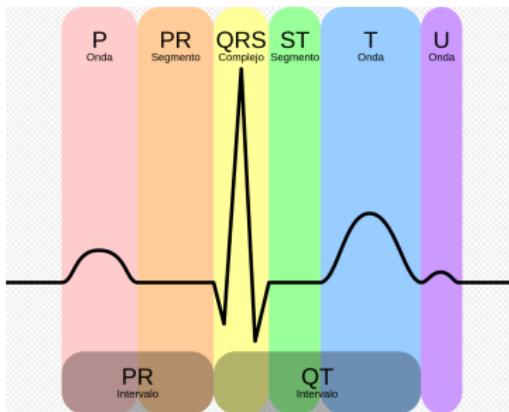
Descripción del problema



- Problema de clasificación de señales obtenidas por ECG.
- Principal objetivo: Discernir entre ritmos normales, FA y otras patologías.

Descripción del problema

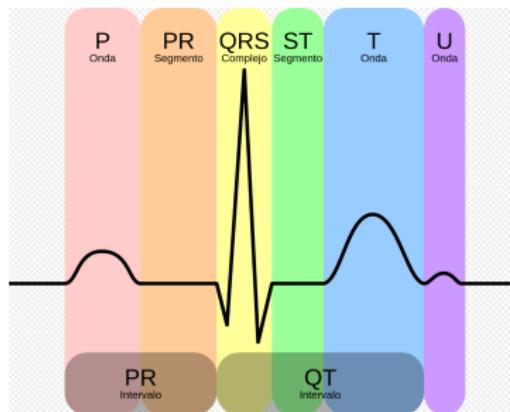
Descripción del problema



- Problema de clasificación de señales obtenidas por ECG.
- Principal objetivo: Discernir entre ritmos normales, FA y otras patologías.
- Existen dos enfoques para abordar esto.

Descripción del problema

Descripción del problema



- Problema de clasificación de señales obtenidas por ECG.
- Principal objetivo: Discernir entre ritmos normales, FA y otras patologías.
- Existen dos enfoques para abordar esto.
 - Enfoque tradicional: Extracción de características a mano
 - Enfoque moderno: *Técnicas de aprendizaje profundo*

Descripción del problema



Base de datos

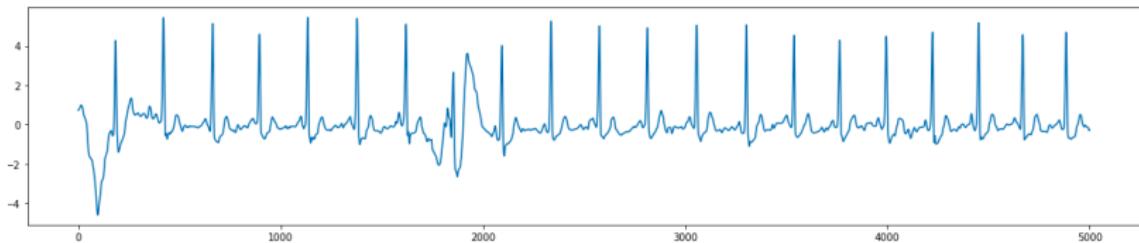
- PhysioNet/CinC, competición 2017.
- 4 clases

Descripción del problema



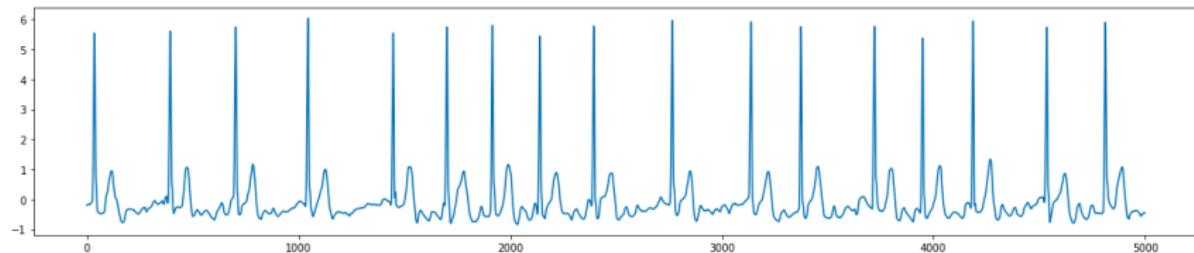
Base de datos

- PhysioNet/CinC, competición 2017.
- 4 clases:
 - Ritmo sinusal normal



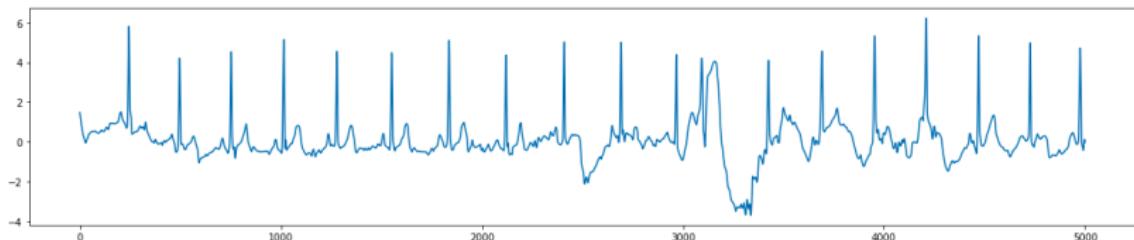
Base de datos

- PhysioNet/CinC, competición 2017.
- 4 clases:
 - Ritmo sinusal normal
 - Fibrilación auricular



Base de datos

- PhysioNet/CinC, competición 2017.
- 4 clases:
 - Ritmo sinusal normal
 - Fibrilación auricular
 - Otro (Otra patología)

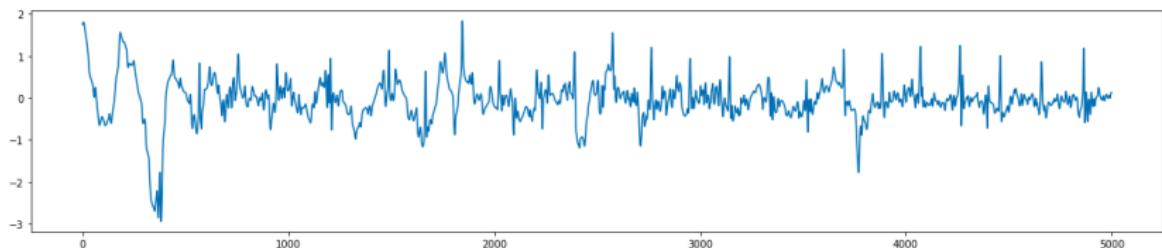


Descripción del problema



Base de datos

- PhysioNet/CinC, competición 2017.
- 4 clases:
 - Ritmo sinusal normal
 - Fibrilación auricular
 - Otro (Otra patología)
 - Ritmo demasiado ruidoso



Base de datos

- PhysioNet/CinC, competición 2017.
- 4 clases:
 - Ritmo sinusal normal
 - Fibrilación auricular
 - Otro (Otra patología)
 - Ritmo demasiado ruidoso
- Fuerte Desbalanceo entre clases

Tipo	Cantidad	Media	SD	Max	Mediana	Min
Normal	5154	31,9	10	61	30	9
AF	771	31,6	12,5	60	30	10
Otro Ritmo	2557	34,1	11,8	60,9	30	9,1
Ruido	46	27,1	9	60	30	10,2
Total	8528	32,5	10,9	61	30	9

Table: Datos del conjunto de entrenamiento

Descripción del problema ○ ○ ○ ● ○ ○ ○

Desarrollo de los modelos

Mejores modelos de la competición

Modus operandis: Extracción de características de manera manual y posterior clasificación.

Modelos:

Descripción del problema ○ ○ ○ ● ○ ○ ○

Desarrollo de los modelos

Mejores modelos de la competición

Modus operandis: Extracción de características de manera manual y posterior clasificación.

Modelos:

- Clasificador binario en cascada

Desarrollo de los modelos

Mejores modelos de la competición

Modus operandis: Extracción de características de manera manual y posterior clasificación.

Modelos:

- Clasificador binario en cascada
- EnCaSe

Desarrollo de los modelos

Mejores modelos de la competición

Modus operandis: Extracción de características de manera manual y posterior clasificación.

Modelos:

- Clasificador binario en cascada
- EnCaSe
- Marco de interpretación abductivo

Desarrollo de los modelos

Mejores modelos de la competición

Modus operandis: Extracción de características de manera manual y posterior clasificación.

Modelos:

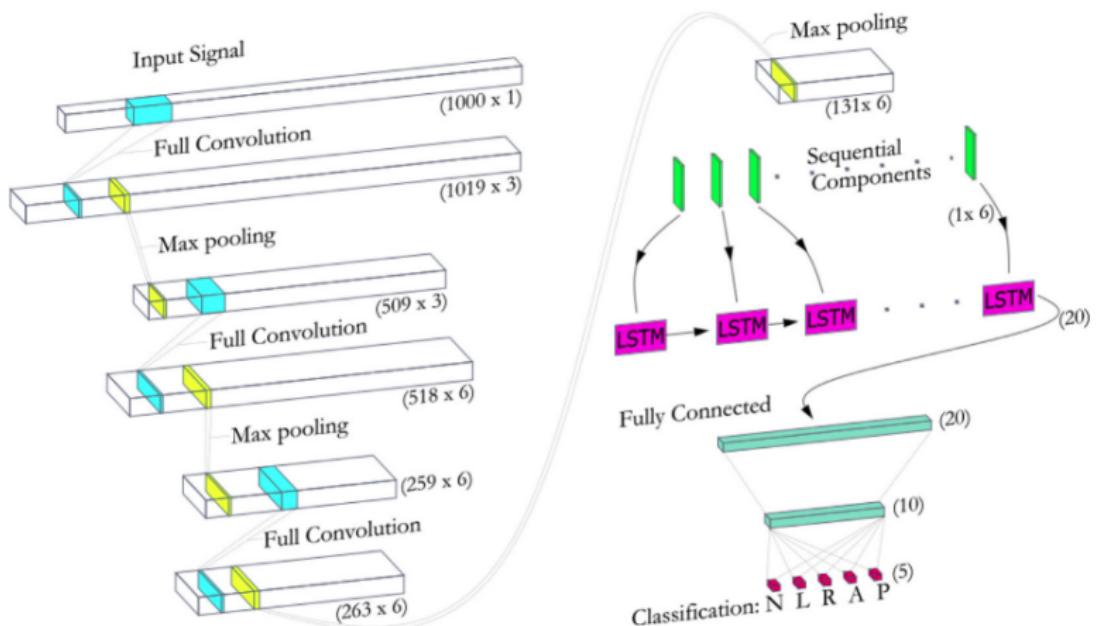
- Clasificador binario en cascada
- EnCaSe
- Marco de interpretación abductivo
- Clasificador basado en Random Forest

Descripción del problema

Desarrollo de los modelos

Modelos propuestos de la literatura

OhShuLi

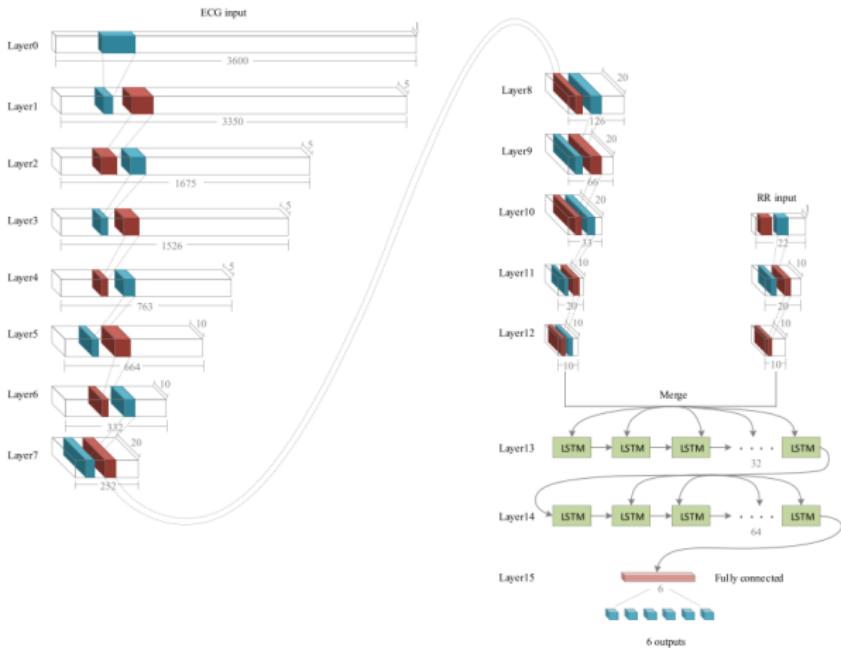


Descripción del problema

Desarrollo de los modelos

Modelos propuestos de la literatura

ChenChen



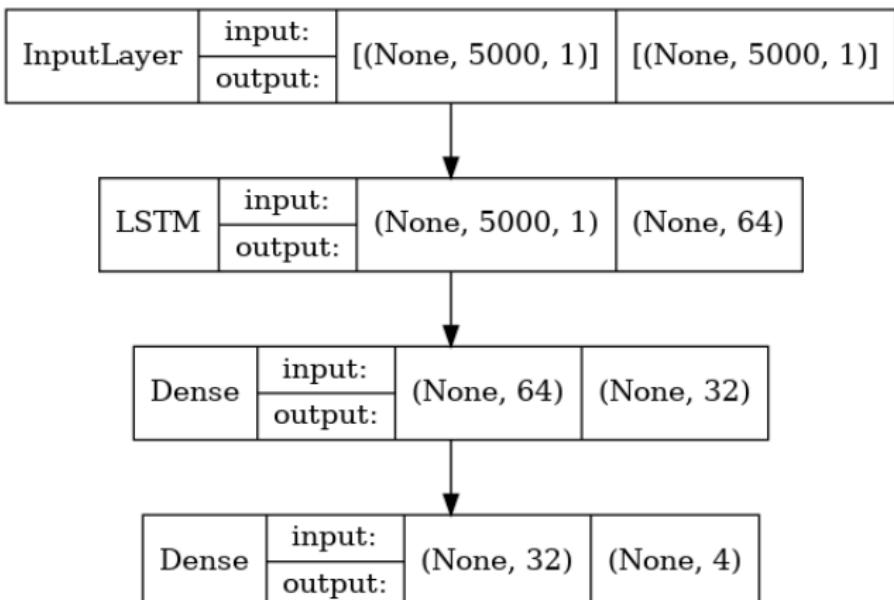
Descripción del problema

○ ○ ○ ○ ● ○ ○

Desarrollo de los modelos

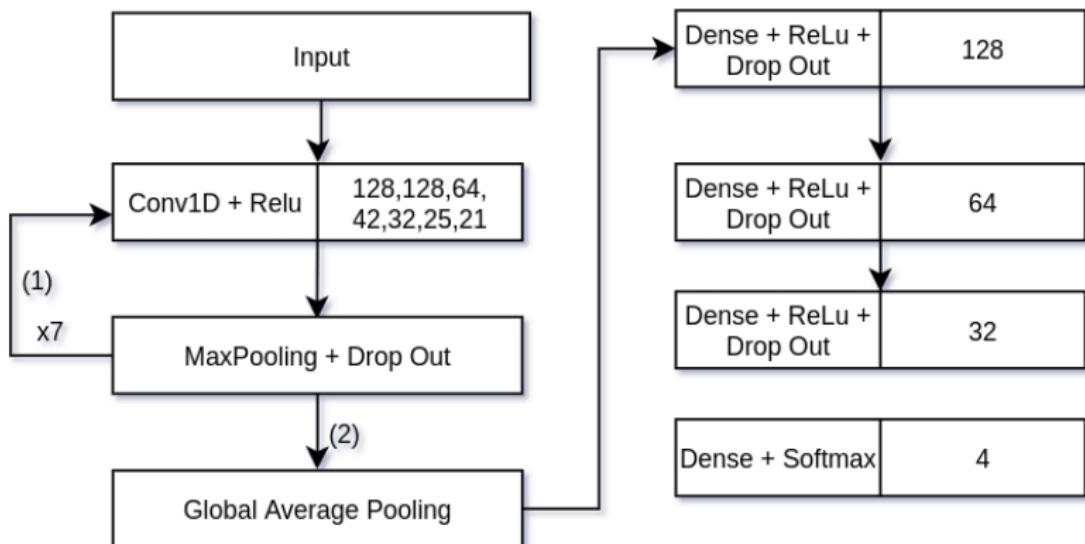
Modelos propuestos de la literatura

GaoJunLi



Modelos propios

CNN



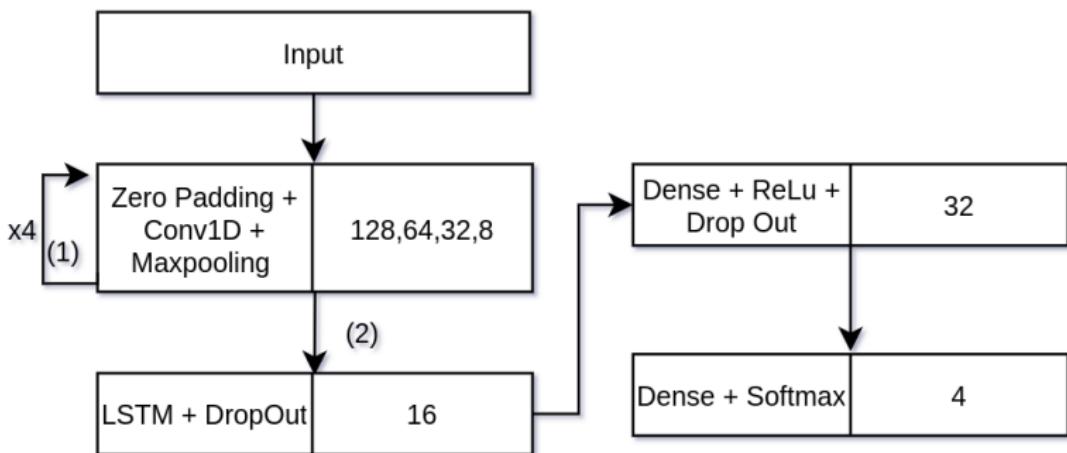
Descripción del problema

○ ○ ○ ○ ○ ● ○

Desarrollo de los modelos

Modelos propios

LSTM



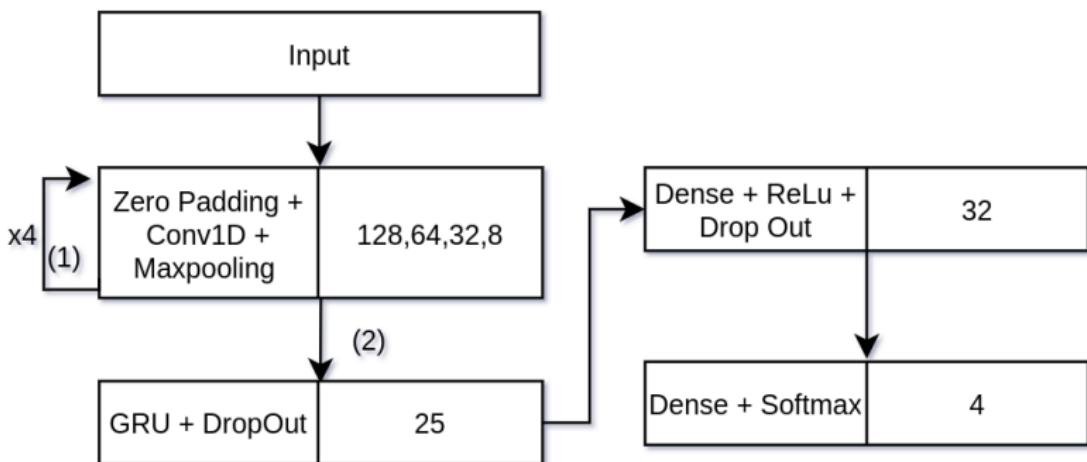
Descripción del problema

○ ○ ○ ○ ○ ● ○

Desarrollo de los modelos

Modelos propios

GRU



Descripción del problema

○ ○ ○ ○ ○ ○ ●

Métrica

F_1 score

		Predicted Classification				
		Normal	AF	Other	Noisy	Total
Reference Classification	Normal	<i>Nn</i>	<i>Na</i>	<i>No</i>	<i>Np</i>	$\sum N$
	AF	<i>An</i>	<i>Aa</i>	<i>Ao</i>	<i>Ap</i>	$\sum A$
	Other	<i>On</i>	<i>Oa</i>	<i>Oo</i>	<i>Op</i>	$\sum O$
	Noisy	<i>Pn</i>	<i>Pa</i>	<i>Po</i>	<i>Pp</i>	$\sum P$
	Total	$\sum n$	$\sum a$	$\sum o$	$\sum p$	

- Ritmo normal:

$$F_{1n} = \frac{2Nn}{\sum N + \sum n}$$

- Ritmo AF:

$$F_{1a} = \frac{2Aa}{\sum N + \sum n}$$

- Otro ritmo:

$$F_{1o} = \frac{2Oo}{\sum O + \sum o}$$

- Ruido:

$$F_{1p} = \frac{2Pp}{\sum P + \sum p}$$

Descripción del problema

○ ○ ○ ○ ○ ○ ●

Métrica

F_1 score

		Predicted Classification				
		Normal	AF	Other	Noisy	Total
Reference Classification	Normal	<i>Nn</i>	<i>Na</i>	<i>No</i>	<i>Np</i>	$\sum N$
	AF	<i>An</i>	<i>Aa</i>	<i>Ao</i>	<i>Ap</i>	$\sum A$
	Other	<i>On</i>	<i>Oa</i>	<i>Oo</i>	<i>Op</i>	$\sum O$
	Noisy	<i>Pn</i>	<i>Pa</i>	<i>Po</i>	<i>Pp</i>	$\sum P$
	Total	$\sum n$	$\sum a$	$\sum o$	$\sum p$	

- F_1 :

$$F_1 = \frac{F_{1n} + F_{1a} + F_{1o} + F_{1p}}{4}$$

Outline

1 Motivation

2 Aprendizaje Estadístico

3 Teorema de Aproximación Universal

4 Redes Neuronales

5 Descripción del problema

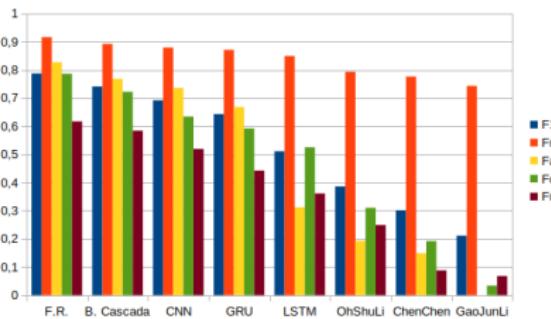
6 Resultados y Análisis

7 Conclusiones

Resultados

Resultados del conjunto de modelos estudiados

	F_n	F_a	F_o	F_r	F₁
B. Cascada	0,916	0,827	0,786	0,617	0,787
R.F.	0,892	0,768	0,722	0,584	0,741
CNN	0,879	0,736	0,634	0,519	0,692
GRU	0,871	0,668	0,592	0,442	0,643
LSTM	0,849	0,312	0,525	0,361	0,511
OhShuLi	0,793	0,193	0,31	0,249	0,386
ChenChen	0,776	0,149	0,192	0,088	0,301
GaoJunLi	0,743	0	0,034	0,068	0,211

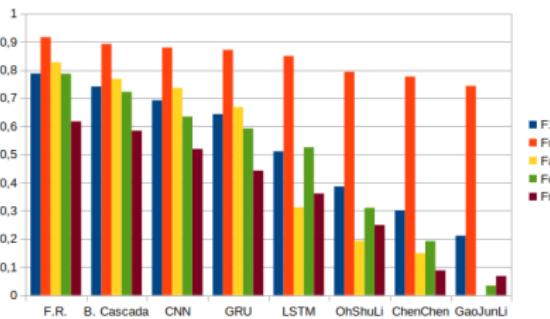


Resultados

Resultados del conjunto de modelos estudiados

	F_n	F_a	F_o	F_r	F₁
B. Cascada	0,916	0,827	0,786	0,617	0,787
R.F.	0,892	0,768	0,722	0,584	0,741
CNN	0,879	0,736	0,634	0,519	0,692
GRU	0,871	0,668	0,592	0,442	0,643
LSTM	0,849	0,312	0,525	0,361	0,511
OhShuLi	0,793	0,193	0,31	0,249	0,386
ChenChen	0,776	0,149	0,192	0,088	0,301
GaoJunLi	0,743	0	0,034	0,068	0,211

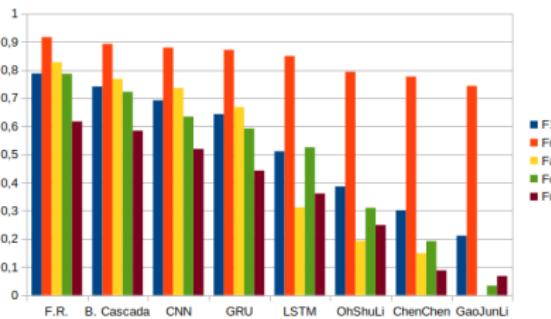
Ranking:



Resultados

Resultados del conjunto de modelos estudiados

	F_n	F_a	F_o	F_r	F₁
B. Cascada	0,916	0,827	0,786	0,617	0,787
R.F.	0,892	0,768	0,722	0,584	0,741
CNN	0,879	0,736	0,634	0,519	0,692
GRU	0,871	0,668	0,592	0,442	0,643
LSTM	0,849	0,312	0,525	0,361	0,511
OhShuLi	0,793	0,193	0,31	0,249	0,386
ChenChen	0,776	0,149	0,192	0,088	0,301
GaoJunLi	0,743	0	0,034	0,068	0,211



Ranking:

- ① Modelos de la competición
- ② Modelos propios
- ③ Propuestas de la literatura

Análisis

Modelos de la competición

Table: Clasf. RF

	Normal	FA	Otro	Ruido
Normal	953	2	57	3
FA	5	116	29	2
Otro	92	10	375	6
Ruido	15	1	10	29

Table: Clasf. Binario en Cascada

	Normal	FA	Otro	Ruido
Normal	962	7	53	6
FA	6	114	28	4
Otro	118	21	333	10
Ruido	15	4	7	31

- Mejores resultados, aunque lejos de la perfección
- Dificultad para distinguir Ritmo normal frente a otras patologías distintas de FA
- Conocimiento a priori.
- Modelos específicos diseñados para el problema

Análisis

Modelos de la literatura

Table: GaoJunLi

	Normal	FA	Otro	Ruido
Normal	1011	0	0	0
FA	147	0	0	0
Otro	491	0	0	0
Ruido	57	0	0	0

Table: OhShuLi

	Normal	FA	Other	Ruido
Normal	980	0	29	0
FA	141	0	6	0
Other	470	0	20	0
Ruido	52	0	5	0

Table: ChenChen

	Normal	FA	Otro	Ruido
Normal	873	2	92	0
FA	122	1	11	0
Otro	431	2	43	0
Ruido	38	0	1	0

- No son capaces de clasificar adecuadamente
- Modelos diseñados para otro problema
- Peores Resultados

Análisis

Modelos propios

Table: CNN

	Normal	FA	Otras	Ruido
Normal	941	6	62	0
FA	117	10	20	0
Otras	371	9	110	0
Ruido	49	2	6	0

Table: GRU

	Normal	FA	Otras	Ruido
Normal	934	6	69	0
FA	114	6	27	0
Otras	375	9	106	0
Ruido	50	1	6	0

Table: LSTM

	Normal	FA	Otras	Ruido
Normal	927	0	82	0
FA	127	0	20	0
Otras	422	0	68	0
Ruido	51	0	6	0

- No son capaces de clasificar la clase minoritaria
- Problemas en distinguir Normal frente a Otro y FA

Outline

- 1 Motivation
- 2 Aprendizaje Estadístico
- 3 Teorema de Aproximación Universal
- 4 Redes Neuronales
- 5 Descripción del problema
- 6 Resultados y Análisis
- 7 Conclusiones

Conclusiones ○ ● ○

Conclusiones

Conclusiones

- La teoría matemática avala el potencial del aprendizaje profundo bajo ciertos supuestos.

Conclusiones

- La teoría matemática avala el potencial del aprendizaje profundo bajo ciertos supuestos.
- *Enfoque Tradicional* vs Enfoque Moderno.

Conclusiones

- La teoría matemática avala el potencial del aprendizaje profundo bajo ciertos supuestos.
- *Enfoque Tradicional* vs Enfoque Moderno.
- Resultados manifiestamente mejorables. Limitaciones:
 - Modelos de la competición → *Normal vs Otro*, conocimiento a priori...
 - Modelos de la literatura → Poca transportabilidad...
 - Modelos propios → Tendencia por la clase mayoritaria...

Conclusiones ○ ○ ●

Trabajos Futuros

Trabajos Futuros

- Mejorar la calidad de las bases de datos (desbalanceo, más ejemplos...)

Trabajos Futuros

- Mejorar la calidad de las bases de datos (desbalanceo, más ejemplos...)
- Mejorar los modelos de la competición

Trabajos Futuros

- Mejorar la calidad de las bases de datos (desbalanceo, más ejemplos...)
- Mejorar los modelos de la competición
- Refinar los modelos propios

Trabajos Futuros

- Mejorar la calidad de las bases de datos (desbalanceo, más ejemplos...)
- Mejorar los modelos de la competición
- Refinar los modelos propios
- Probar nuevas técnicas: ensemble, stacking, boosting...

Trabajos Futuros

- Mejorar la calidad de las bases de datos (desbalanceo, más ejemplos...)
- Mejorar los modelos de la competición
- Refinar los modelos propios
- Probar nuevas técnicas: ensemble, stacking, boosting...
- Aumentar el número de clases a clasificar

Muchas gracias por la atención

Hasta la próxima :)