Workshop en Ciencias de la Computación

Regularización con cotas de Lipschitz: Reduciendo el sobreajuste en redes neuronales

Manuela Viviana Chacón Chamorro

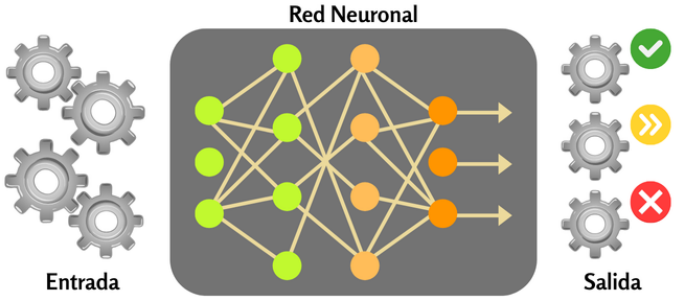
Candidata a Doctora en Ingeniería
Universidad de los Andes

Manizales, Noviembre 26, 2025

Motivación

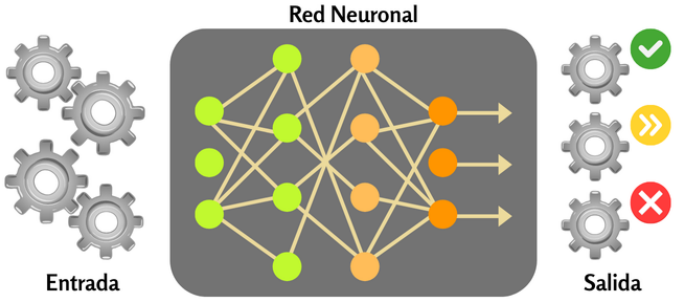
Motivación

- **Meta:** Inspección de calidad en una línea de producción.
- **Solución:** Entrenar una red neuronal para detectar defectos superficiales en piezas metálicas.



Motivación

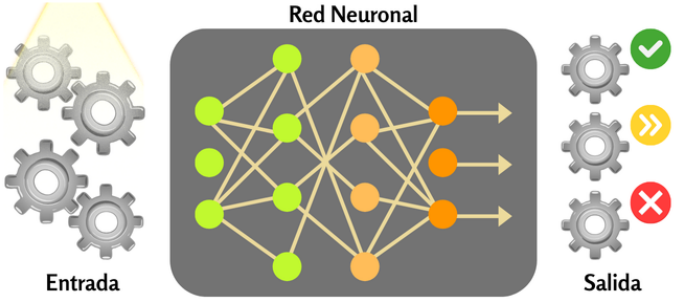
- **Meta:** Inspección de calidad en una línea de producción.
- **Solución:** Entrenar una red neuronal para detectar defectos superficiales en piezas metálicas.



En validación interna, el modelo alcanza una precisión excelente. Pero al pasar a producción, el rendimiento cae de forma inesperada.

Motivación

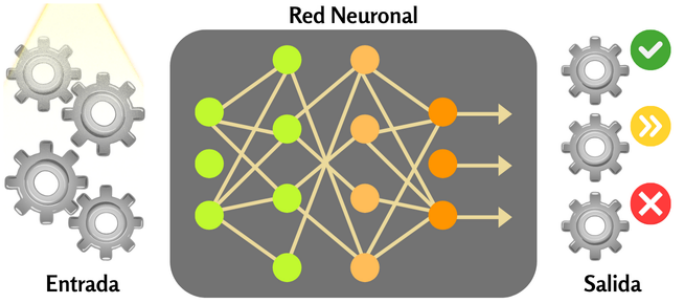
- **Meta:** Inspección de calidad en una línea de producción.
- **Solución:** Entrenar una red neuronal para detectar defectos superficiales en piezas metálicas.



Una variación en el tono y la dirección de la luz hace que la red comience a confundir texturas normales con defectos, generando rechazos innecesarios y retrasos en toda la cadena.

Motivación

- **Meta:** Inspección de calidad en una línea de producción.
- **Solución:** Entrenar una red neuronal para detectar defectos superficiales en piezas metálicas.



Modelo demasiado sensible: había aprendido fronteras de decisión muy irregulares, adaptadas milimétricamente a los ejemplos de entrenamiento

Motivación

Sobre-ajuste / Overfitting

Una red excesivamente sensible puede ajustarse casi perfectamente a los datos de entrenamiento, pero ante pequeñas variaciones en las entradas reales deja de funcionar bien fuera de ese conjunto.

Motivación

Sobre-ajuste / Overfitting

Una red excesivamente sensible puede ajustarse casi perfectamente a los datos de entrenamiento, pero ante pequeñas variaciones en las entradas reales deja de funcionar bien fuera de ese conjunto.

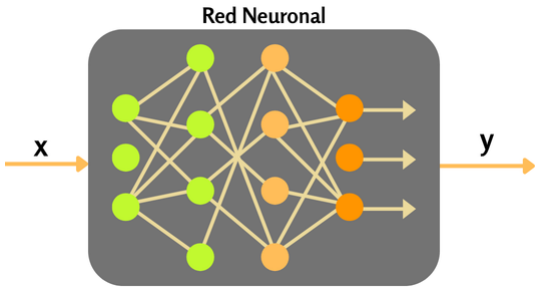
El contenido de esta charla está basado en el artículo: M. Chacon-Chamorro, F. A. Gallego, J. C. Riaño-Rojas, *Reducing overfitting in ResNet with Adaptive Lipschitz regularization*, Journal of Computational and Applied Mathematics, 116747, 2025.

Outline

1. Motivación
2. Elementos conceptuales
3. Método LBA
4. Experimentos y Resultados
5. Conclusiones

Elementos conceptuales

¿Qué es una red neuronal?

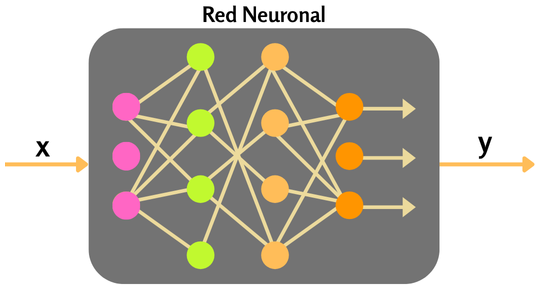


- Una red neuronal es un conjunto de capas de neuronas conectadas que implementan una composición de funciones.
- Cada neurona calcula una operación del tipo

$$a = \sigma(w^\top z + b),$$

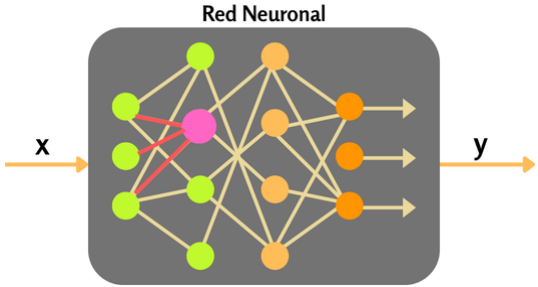
donde z es el vector de salidas de la capa anterior, w son los pesos de las aristas incidentes y b es un sesgo.

¿Qué es una red neuronal?



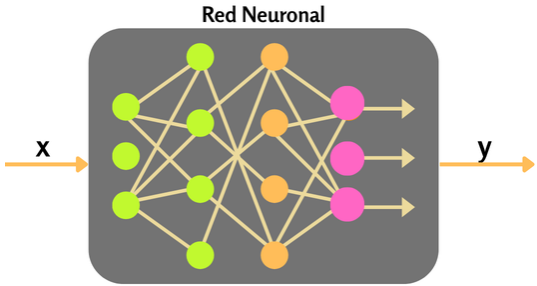
- El vector de datos de entrada x se alimenta en la primera capa.

¿Qué es una red neuronal?



- En cada capa, las neuronas combinan las entradas mediante los pesos y aplican la función de activación σ.

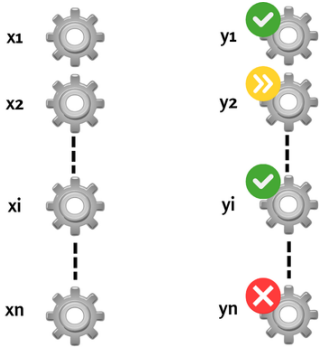
¿Qué es una red neuronal?



- Las activaciones se propagan capa a capa hasta obtener la salida del modelo.

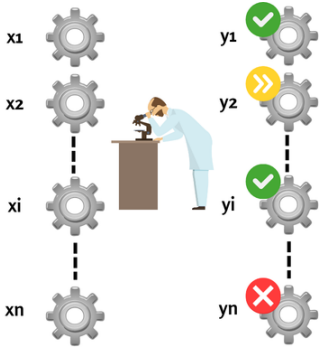
¿Cómo aprende la red neuronal?

- Una red neuronal aprende a partir de ejemplos $\{(\mathrm{x}_i, y_i)\}_{i=1}^{N}$.



¿Cómo aprende la red neuronal?

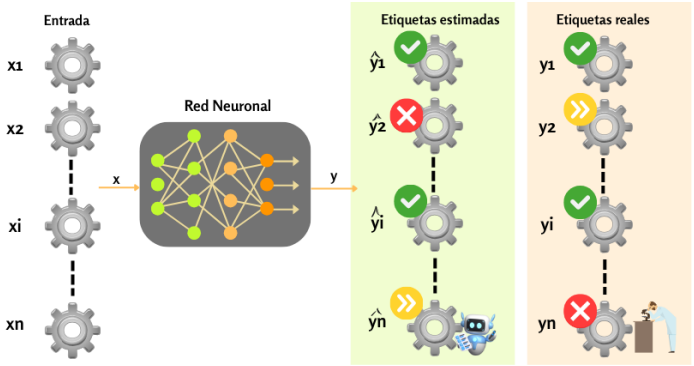
- Una red neuronal aprende a partir de ejemplos $\{(\mathrm{x}_i, y_i)\}_{i=1}^{N}$.



Y. Bengio, I. Goodfellow, A. Courville, Deep Learning, MIT Press, Cambridge, MA, USA (2017)

¿Cómo aprende la red neuronal?

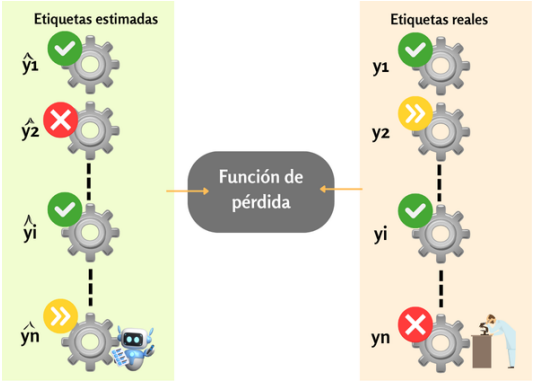
- Para cada entrada x_i, la red produce una salida $\hat{y}_i = f_\theta(\mathrm{x}_i)$, donde θ representa todos los pesos del modelo.



Y. Bengio, I. Goodfellow, A. Courville, Deep Learning, MIT Press, Cambridge, MA, USA (2017)

¿Cómo aprende la red neuronal?

- Definimos una función de pérdida $\mathcal{L}(y_i, \hat{y}_i)$ que mide el error entre la predicción y la etiqueta real.



Y. Bengio, I. Goodfellow, A. Courville, Deep Learning, MIT Press, Cambridge, MA, USA (2017)

¿Cómo aprende la red neuronal?

- El aprendizaje consiste en resolver el problema de optimización

$$\min_{\theta} \quad \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f_\theta(\mathrm{x}_i)).$$

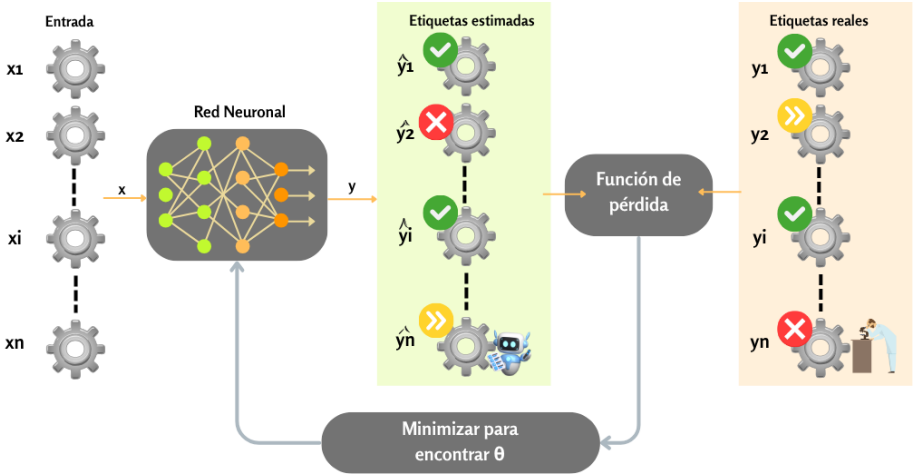
- El algoritmo ajusta los pesos usando gradientes para reducir la pérdida y mejorar la capacidad de predicción.

Aprender

Ajustar los parámetros para que la red se acerque cada vez más a los ejemplos reales.

Y. Bengio, I. Goodfellow, A. Courville, Deep Learning, MIT Press, Cambridge, MA, USA (2017)

Esquema de aprendizaje



¿Qué es el sobre-ajuste?

Es la incapacidad del modelo para generalizar. El modelo funciona bien con los datos que se usaron en el entrenamiento $\{(x_i, y_i)\}$, pero falla con datos en un escenario de producción.

¿Qué es el sobre-ajuste?

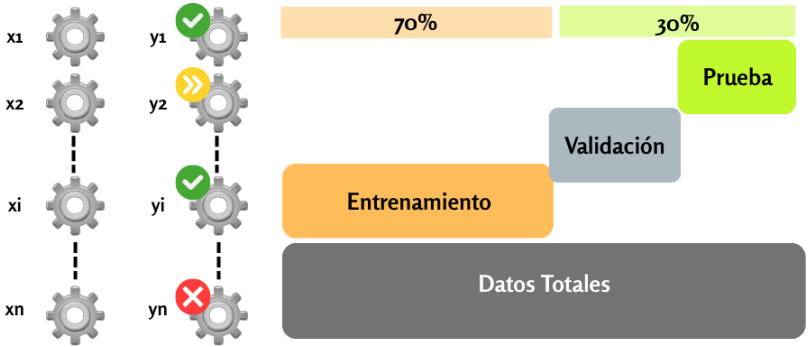
Es la incapacidad del modelo para generalizar. El modelo funciona bien con los datos que se usaron en el entrenamiento $\{(x_i, y_i)\}$, pero falla con datos en un escenario de producción.

- Un modelo sobreajustado ha aprendido tanto los detalles y el ruido del conjunto de entrenamiento que no puede generalizar bien a nuevos datos.
- Modelos ineficiente en entornos operativos.
- Falsas expectativas de rendimiento.
- El sobre-ajuste puede resultar en modelos innecesariamente complejos que requieren más recursos computacionales.

S. Aburass and M. Abu Rumman, "Quantifying overfitting: introducing the overfitting index," Proc. 2024 Int. Conf. on Electrical, Computer and Energy Technologies (ICECET), pp. 1–7, 2024.
Li, H., G. K. Rajbahadur, D. Lin, C.-P. Bezemer, and Z. M. Jiang, "Keeping deep learning models in check: A history-based approach to mitigate overfitting," *IEEE Access*, vol. 12, pp. 70676–70689, 2024.

¿Cómo se manifiesta el sobre-ajuste?

- Desempeño adecuado en el entrenamiento, pero bajo en validación.



¿Cómo se manifiesta el sobre-ajuste?

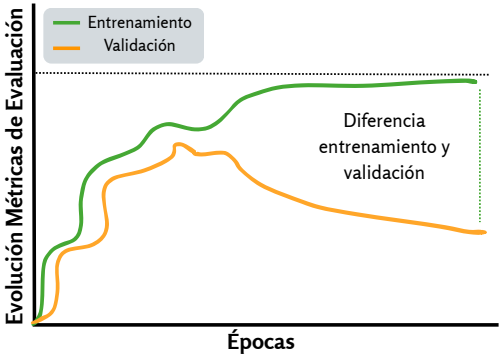
- Desempeño adecuado en el entrenamiento, pero bajo en validación.
- Sensibilidad a cambios en los datos de entrenamiento.

¿Cómo se manifiesta el sobre-ajuste?

- Desempeño adecuado en el entrenamiento, pero bajo en validación.
- Sensibilidad a cambios en los datos de entrenamiento.
- Alta varianza en el desempeño.

¿Cómo se manifiesta el sobre-ajuste?

- Curvas de aprendizaje divergentes.



¿Cuál es el origen?

¿Cuál es el origen?

En los datos

- Tamaño de los datos
- Distribución de los datos
- Falta de variabilidad
- Incorrecto tratamiento de los datos

¿Cuál es el origen?

En los datos

- Tamaño de los datos
- Distribución de los datos
- Falta de variabilidad
- Incorrecto tratamiento de los datos

En el modelo

- **Complejidad del modelo**
- Hiperparámetros mal ajustados

B. Dherin, M. Munn, M. Rosca, and D. Barrett, "Why neural networks find simple solutions: The many regularizers of geometric complexity," Advances in Neural Information Processing Systems, vol. 35, pp. 2333–2349, 2022.
G. Khromov and S. P. Singh, "Some fundamental aspects about Lipschitz continuity of neural networks," arXiv preprint arXiv:2302.10886, 2023

Método LBA

Proceso de regularización

La filosofía de las regularizaciones es restringir la complejidad del modelo, se considera que un modelo sobre parametrizado podría conducir al sobreajuste. En este caso se modifica la función de costo agregando un parámetro que penalice la complejidad del modelo

Proceso de regularización

La filosofía de las regularizaciones es restringir la complejidad del modelo, se considera que un modelo sobre parametrizado podría conducir al sobreajuste. En este caso se modifica la función de costo agregando un parámetro que penalice la complejidad del modelo

$$\mathcal{L} = \text{Error de clasificación} + \lambda \text{Complejidad del modelo}$$

Proceso de regularización

La filosofía de las regularizaciones es restringir la complejidad del modelo, se considera que un modelo sobre parametrizado podría conducir al sobreajuste. En este caso se modifica la función de costo agregando un parámetro que penalice la complejidad del modelo

$$\mathcal{L} = \text{Error de clasificación} + \lambda \text{Complejidad del modelo}$$

El proceso de regularización es una buena técnica para mitigar el sobre-ajuste, sin embargo, agrega un nuevo hiperparámetro para sintonizar λ y requiere una forma de medir la complejidad del modelo.

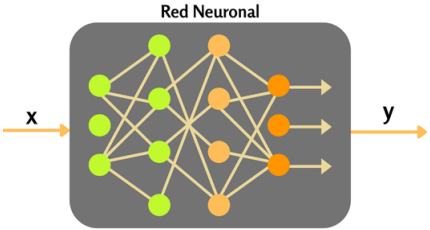
C. F. G. dos Santos and J. P. Papa, "Avoiding overfitting: A survey on regularization methods for convolutional neural networks," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–25, 2022.
H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of neural networks by enforcing Lipschitz continuity," *Machine Learning*, vol. 110, no. 2, pp. 393–416, 2021.

Medir la complejidad del modelo

La complejidad del modelo está sujeta a una métrica la cual suele ser relacionada con una norma de los pesos de las conexiones.

Medir la complejidad del modelo

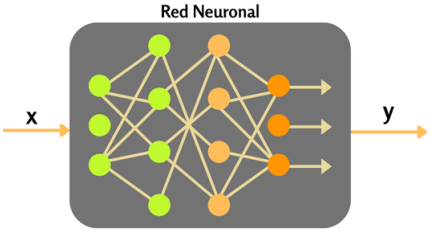
La complejidad del modelo está sujeta a una métrica la cual suele ser relacionada con una norma de los pesos de las conexiones.



$$||\theta||_p$$

Medir la complejidad del modelo

La complejidad del modelo está sujeta a una métrica la cual suele ser relacionada con una norma de los pesos de las conexiones.



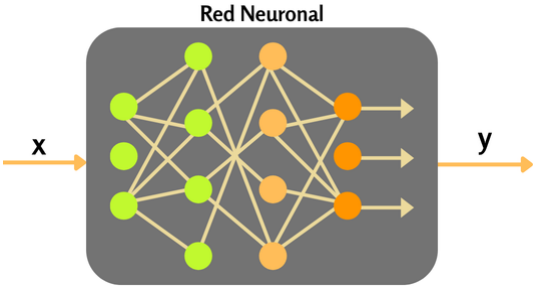
$$||\theta||_p$$

Teorema (Equivalencia de normas en $\mathbb{R}^n$)

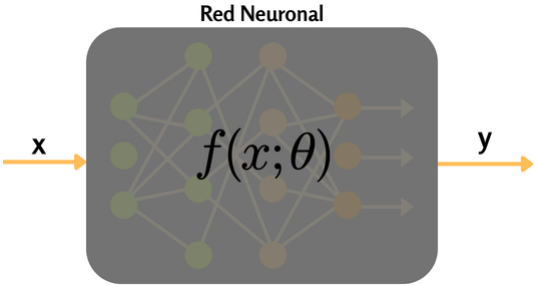
Para cualquier par de normas $\|\cdot\|_a$ y $\|\cdot\|_b$ en $\mathbb{R}^n$, existen constantes $c_1, c_2 > 0$ tales que, para todo $\theta \in \mathbb{R}^n$,

$$c_1\|\theta\|_a \;\leq\; \|\theta\|_b \;\leq\; c_2\|\theta\|_a.$$

¿Cómo más podemos pensar el sobre-ajuste?

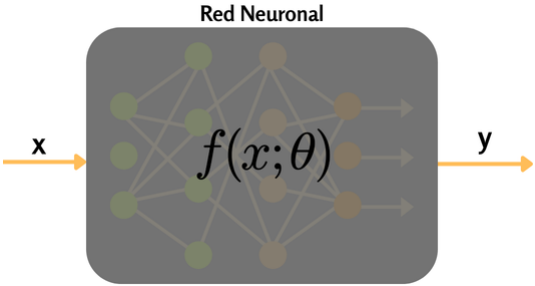


¿Cómo más podemos pensar el sobre-ajuste?



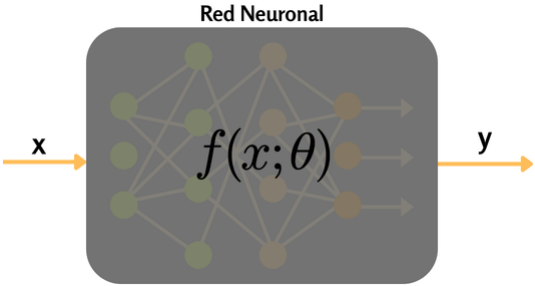
El modelo neuronal puede interpretarse como una función, en el sentido matemático, esta función mapea del espacio de las características al espacio de las etiquetas.

Propiedades modelo neuronal



- Si el modelo neuronal es una función entonces podemos plantear o analizar sobre la función algunas propiedades matemáticas.

Propiedades modelo neuronal



- Si el modelo neuronal es una función entonces podemos plantear o analizar sobre la función algunas propiedades matemáticas.
- Una función es **Lipschitz continua** si los cambios en sus valores de salida son proporcionales a los cambios en sus entradas, sin que haya saltos bruscos.

Continuidad de Lipschitz

$$||f(x_i;\theta) - f(x_j;\theta)|| \leq LB||x_i - x_j||$$

Teorema

La función que representa una arquitectura neuronal convencional es Lipschitz continua si las funciones de activación de las neuronas también lo son. Típicamente estas funciones son Lipschitz continuas.

Continuidad de Lipschitz

$$||f(x_i;\theta) - f(x_j;\theta)|| \leq LB||x_i - x_j||$$

Teorema

La función que representa una arquitectura neuronal convencional es Lipschitz continua si las funciones de activación de las neuronas también lo son. Típicamente estas funciones son Lipschitz continuas.

¿Cómo debería ser LB para no sobre-ajustar?

Core Idea

La constante LB debería ser "pequeña" o cercana a uno para evitar que en el espacio de salida a cambios pequeños en la entrada refleje un cambio abrupto en la salida (cambiar de etiqueta).

¿De qué depende LB en el modelo neuronal?

$$LB = \prod_{i=1}^{L} \sqrt{\rho(W^{(i)\top} W^{(i)})}$$

Idea del método

- Usar la constante LB como valor λ en la regularización.

$$\mathcal{L}_r = \mathcal{L}(\theta) + \lambda||\theta||_2$$

$$\mathcal{L}_r = \mathcal{L}(\theta) + LB||\theta||_2$$

- Usar aleatoriamente una capa y usar su valor LB como penalización en la regularización. Esto permitirá que las otras capas "aprendan" mientras se realiza el proceso de regularización.

$$\mathcal{L}_r = \mathcal{L}(\theta) + LB^{(a)}||\theta^{(a)}||_2$$

Experimentos y Resultados

Conjuntos de datos

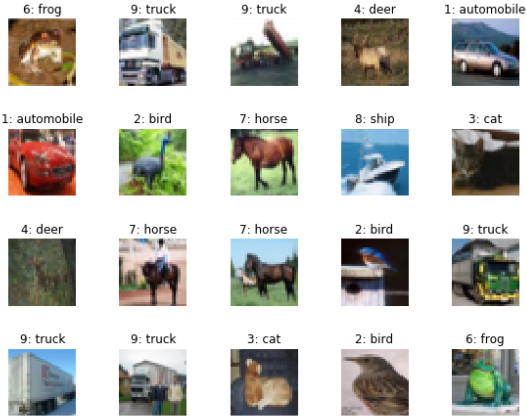
- Imágenes de dígitos MNIST

Conjuntos de datos

- Imágenes de prendas de vestir MNIST

Conjuntos de datos

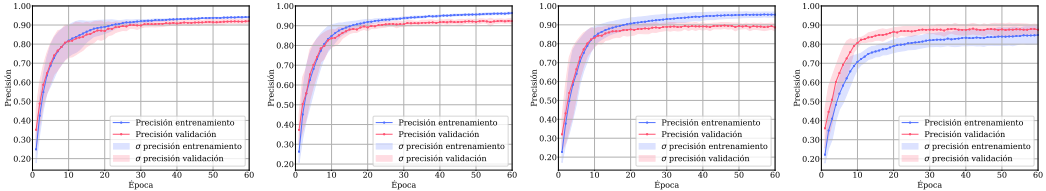
- CIFAR-10



Conjuntos de datos

- Imágenes de dígitos MNIST
- Imágenes de prendas de vestir MNIST
- CIFAR-10
- Datos sintéticos
- Conjuntos de datos predefinidos

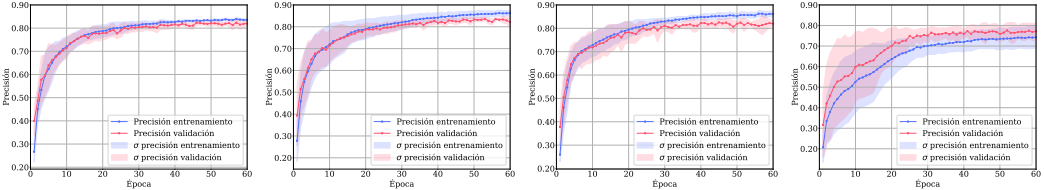
Resultados del método



(a) Regularización LBA (b) Regularización L2 (c) Regularización L1 (d) *Dropout*

Curvas de aprendizaje modelo de 8 capas residuales. Conjunto de datos MNIST.

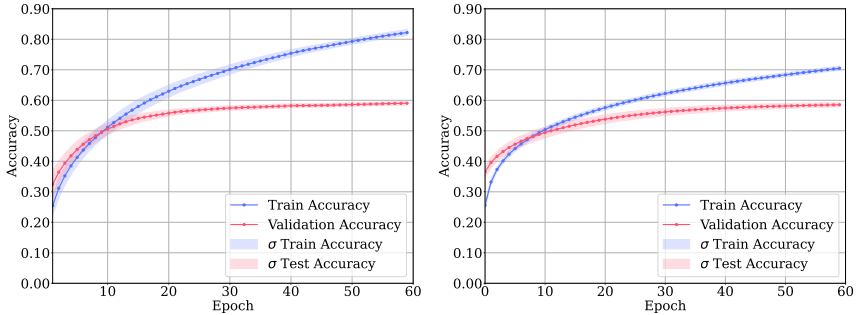
Resultados del método



(a) Regularización LBA (b) Regularización L2 (c) Regularización L1 (d) *Dropout*

Curvas de aprendizaje modelo de 8 capas residuales. Conjunto de datos Fashion MNIST.

Resultados del método

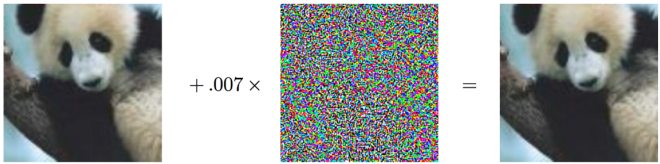


(a) Baseline (b) Regularización LBA

Curvas de aprendizaje modelo CNN. Conjunto de datos CIFAR-10.

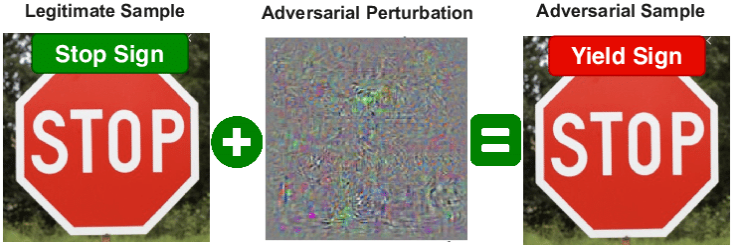
Imágenes adversarias

Una imagen adversaria es el resultado de modificar sutilmente una imagen original, sin embargo, pese a que la modificación es imperceptible para el ojo humano, estas imágenes tienden a "confundir" al algoritmo de clasificación, en particular cuando este se encuentra sobre-ajustado.



Imágenes adversarias

Una imagen adversaria es el resultado de modificar sutilmente una imagen original, sin embargo, pese a que la modificación es imperceptible para el ojo humano, estas imágenes tienden a "confundir" al algoritmo de clasificación, en particular cuando este se encuentra sobre-ajustado.



I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.
N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14, 2017.

Defensa adversaria

Metrics	$\bar{A}_{train}$	$\bar{A}_{test}$	FGSM 1	FGSM 2	FGSM 3	PGD
Model	**3 Layers**					
Method	Original		Adversarial			
Baseline	88.48	83.12	63.60	40.50	19.80	41.90
LBA	79.13	77.68	73.70	67.30	61.30	77.20
L2	87.88	83.48	60.10	49.00	29.80	47.70
L1	88.70	83.61	68.30	46.60	26.30	37.70
Dropout	81.41	83.03	71.70	58.10	45.80	56.50
Model	**5 Layers**					
Method	Original		Adversarial			
Baseline	87.49	79.92	60.70	32.70	16.20	34.30
LBA	81.46	80.28	74.30	66.80	54.60	74.30
L2	87.52	82.84	66.80	43.10	26.60	35.20
L1	86.74	83.50	74.40	60.10	46.90	62.20
Dropout	80.02	80.06	66.20	52.40	42.30	49.40
Model	**7 Layers**					
Method	Original		Adversarial			
Baseline	89.76	82.82	63.30	38.70	23.50	28.00
LBA	83.81	80.28	75.30	65.20	58.30	74.30
L2	88.85	83.29	65.20	42.20	25.30	62.30
L1	84.94	80.67	61.20	38.40	25.90	65.60
Dropout	77.31	78.19	60.30	46.80	36.70	46.50

Conclusiones

Conclusiones

- El sobre-ajuste sigue siendo un problema central en redes neuronales y un desafío abierto en aplicaciones reales.
- Analizar la red como una función permite introducir herramientas matemáticas que ayudan a entender y controlar su comportamiento.
- El método propuesto, basado en cotas de Lipschitz, reduce de manera consistente el sobre-ajuste en arquitecturas sencillas.
- La misma regularización proporciona, además, una defensa efectiva ante perturbaciones adversariales en imágenes.

Agradecimientos

A la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de Colombia, Sede Manizales, por la invitación al Workshop en Ciencias de la Computación, en particular al profesor Fabián Serrano por la gestión de mi participación y, de manera muy especial, al profesor Juan Carlos Riaño por la invitación.

El trabajo presentado corresponde a la divulgación de resultados de mi tesis de maestría y del artículo: "*Reducing overfitting in ResNet with Adaptive Lipschitz regularization*" , Journal of Computational and Applied Mathematics, en coautoría con los profesores Fernando Gallego y Juan Carlos Riaño.

Gracias

¿Preguntas?