

1a
Emisión

Módulo 8

Introducción al Deep Learning

Redes neuronales profundas

Instructor: Gibran Fuentes Pineda



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Dirección General de Cómputo y de Tecnologías de información y Comunicación
Dirección de Docencia en TIC



Educación
Continua
1971 - 2021

Objetivo del tema



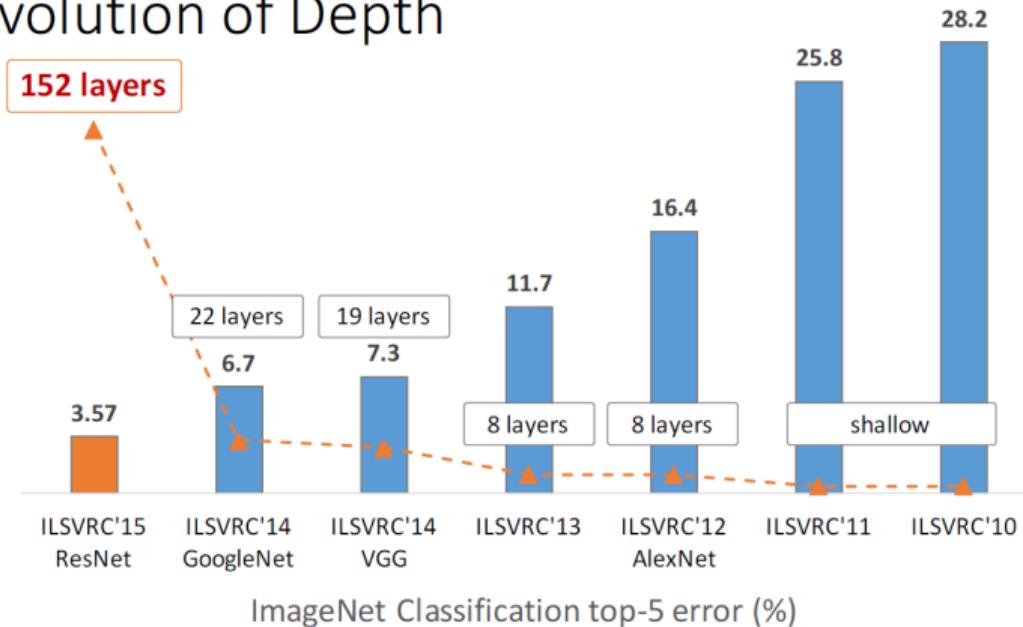
- ▶ El participante identificará las características de las redes neuronales profundas y las técnicas para evitar el sobreajuste y mejorar su entrenamiento.

Sub-temas

- 5.1 Motivación de redes neuronales profundas
- 5.2 Sobreajuste y regularización
- 5.3 Explosión y desvanecimiento del gradiente
- 5.4 Capas de normalización

Motivación de redes neuronales profundas

Revolution of Depth



Sobreajuste y regularización: estrategias de regularización

- ▶ Estrategias para reducir sobreajuste
 - ▶ Penalización de función de función de pérdida
 - ▶ Paro temprano
 - ▶ Aprendizaje de múltiples tareas
 - ▶ Ensamblés
 - ▶ Adición de ruido a entradas, salidas y/o parámetros
 - ▶ Dropout
 - ▶ Acrecentamiento de datos
 - ▶ Normalización por lotes



Sobreajuste y regularización: penalización por norma ℓ_1 y ℓ_2

- ▶ Norma ℓ_1

$$\tilde{E}(\boldsymbol{\theta}) = - \sum_{i=1}^n \{y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\} + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_1$$

- ▶ Norma ℓ_2

$$\tilde{E}(\boldsymbol{\theta}) = - \sum_{i=1}^n \{y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\} + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

Sobreajuste y regularización: paro temprano

- ▶ Se detiene entrenamiento si pérdida o métrica de validación no aumenta después de varios pasos
- ▶ Usualmente se elige el modelo con mejor desempeño en el conjunto de validación

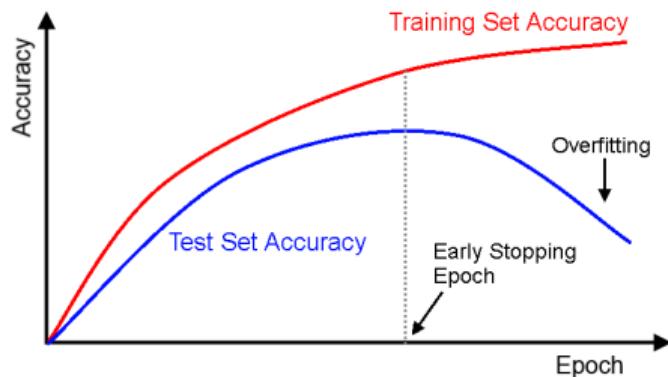


Imagen tomada de <https://deeplearning4j.org/earlystopping>

Sobreajuste y regularización: aprendizaje multitarea

- ▶ Tener una representación genérica compartida entre 2 tareas relacionadas

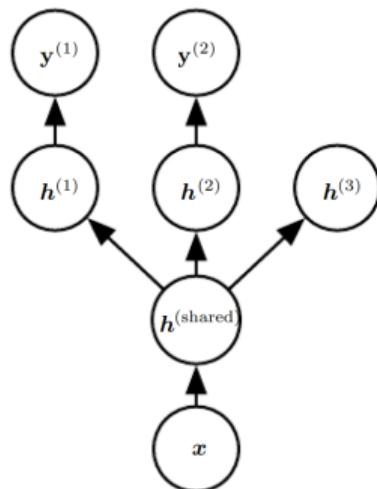
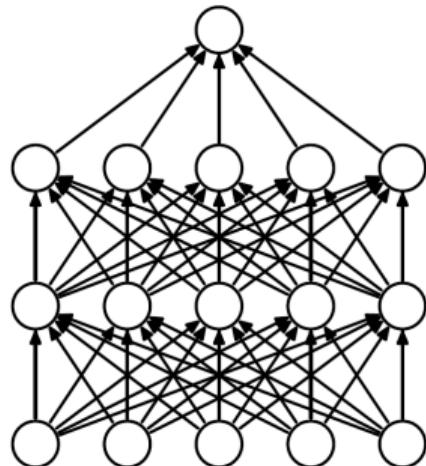


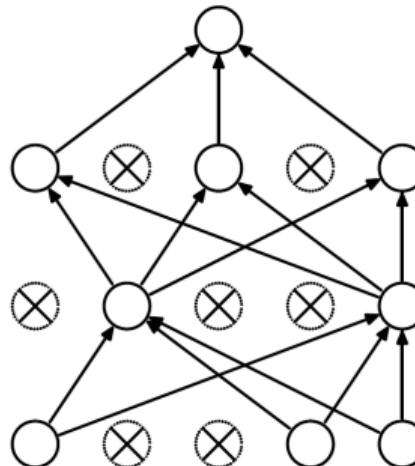
Imagen tomada de Goodfellow et al. Deep Learning, 2016

Sobreajuste y regularización: Dropout (desactivación)

- Desactiva neuronas de forma aleatoria ¹ para evitar co-adaptación



(a) Standard Neural Net



(b) After applying dropout.

Imagen tomada de Srivastava et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, 2014

¹Probabilidad es típicamente 0.5

Sobreajuste y regularización: Dropout en entrenamiento

- ▶ La salida de la i -ésima neurona está dada por

$$\begin{aligned} z_i^{\{\ell+1\}} &= \mathbf{w}_i^{\{\ell+1\}} \tilde{\mathbf{y}}^{\{\ell\}} + b_i^{\{\ell+1\}} \\ y_i^{\{\ell+1\}} &= \phi(z_i^{\{\ell+1\}}) \end{aligned}$$

donde $\tilde{\mathbf{y}}$ es una máscara binaria sobre las salidas de las neuronas con 1s para las activas y 0s para las inactivas

$$\begin{aligned} r_j &\sim \text{Bernoulli}(P) \\ \tilde{\mathbf{y}}^{\{\ell\}} &= \mathbf{r}^{\{\ell\}} * \mathbf{y}^{\{\ell\}} \end{aligned}$$

- ▶ P es un hiperparámetro que indica la probabilidad de que una neurona se mantenga activa

Sobreajuste y regularización: Dropout en inferencia

- ▶ En vez de promediar las salidas de todas las redes entrenadas, se obtiene la salida de una sola red con los pesos y sesgos ($\theta = \{\mathbf{W}, \mathbf{b}\}$) escalados

$$\theta_{\text{inferencia}} = P \cdot \theta$$

- ▶ De esta forma se combinan 2^n redes en una sola

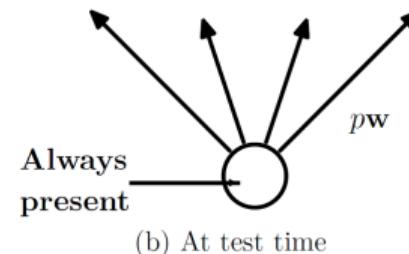
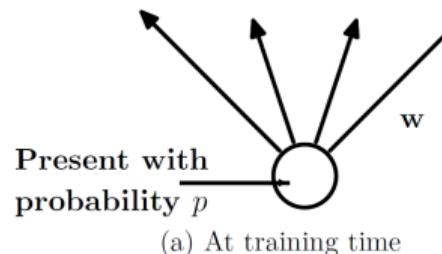


Imagen tomada de Srivastava et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, 2014

Explosión y desvanecimiento del gradiente

- ▶ Problemas con el desvanecimiento y explosión de respuestas (hacia adelante) y gradientes (hacia atrás)

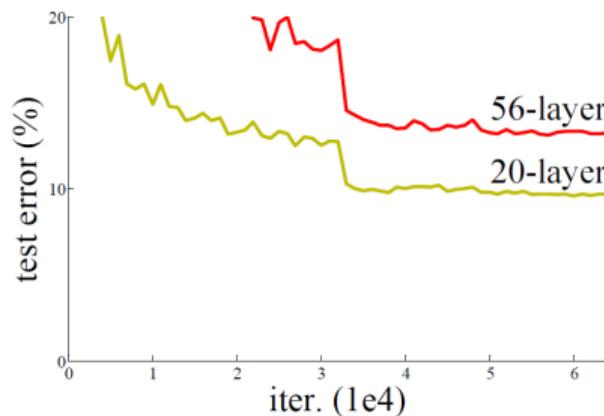
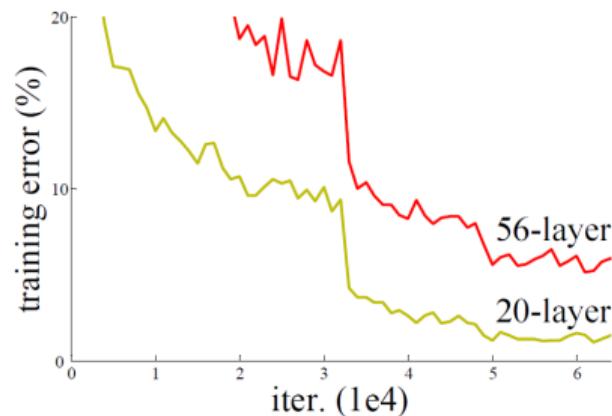
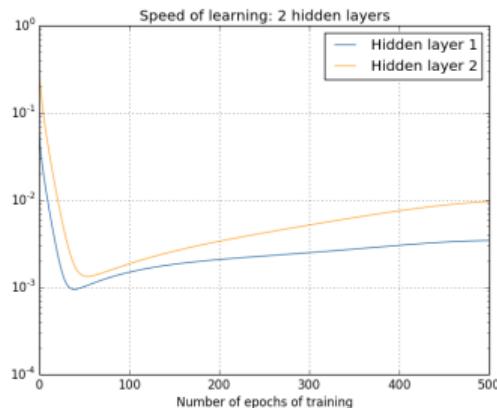


Imagen tomada de He et al. *Deep Residual Learning for Image Recognition*, 2015

Explosión y desvanecimiento del gradiente: 2 capas ocultas

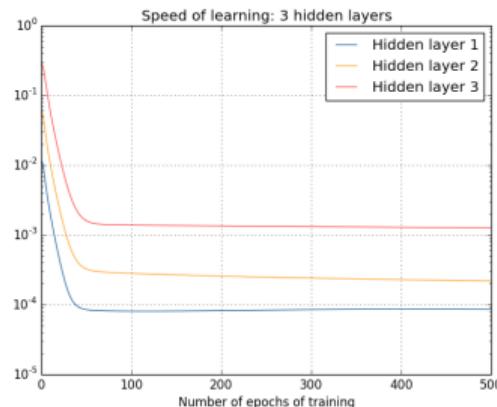
- ▶ Gradientes de primeras capas se vuelven muy pequeños si la red es muy profunda
- ▶ Muy lento actualizar pesos de estas capas



Tomado de <http://neuralnetworksanddeeplearning.com/chap5.html>

Explosión y desvanecimiento del gradiente: 3 capas ocultas

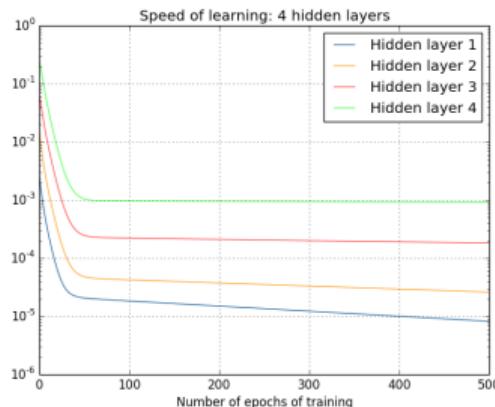
- ▶ Gradientes de primeras capas se vuelven muy pequeños si la red es muy profunda
- ▶ Muy lento actualizar pesos de estas capas



Tomado de <http://neuralnetworksanddeeplearning.com/chap5.html>

Explosión y desvanecimiento del gradiente: 4 capas ocultas

- ▶ Gradientes de primeras capas se vuelven muy pequeños si la red es muy profunda
- ▶ Muy lento actualizar pesos de estas capas

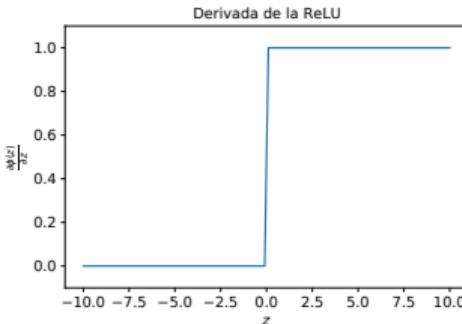
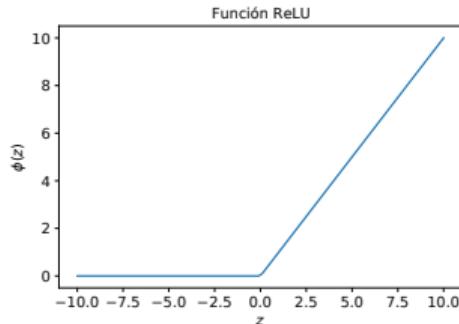
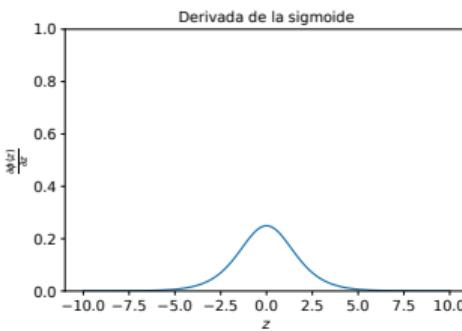
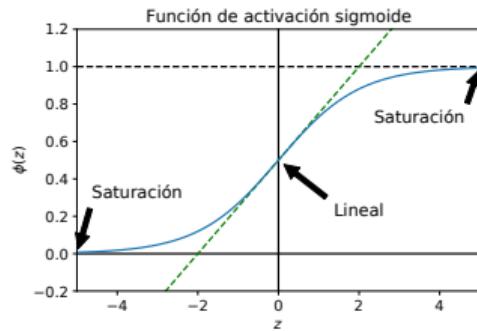


Tomado de <http://neuralnetworksanddeeplearning.com/chap5.html>

Explosión y desvanecimiento del gradiente: mitigación

- ▶ Emplear funciones de activación no saturadas en capas ocultas
- ▶ Incorporar conexiones residuales a la red
- ▶ Inicializar pesos y sesgos con heurísticas apropiadas
- ▶ Recortar los gradientes
- ▶ Emplear normalización por lotes

Explosión y desvanecimiento del gradiente: función de activación



Explosión y desvanecimiento del gradiente: conexiones residuales

- ▶ Agregando conexiones residuales en la arquitectura

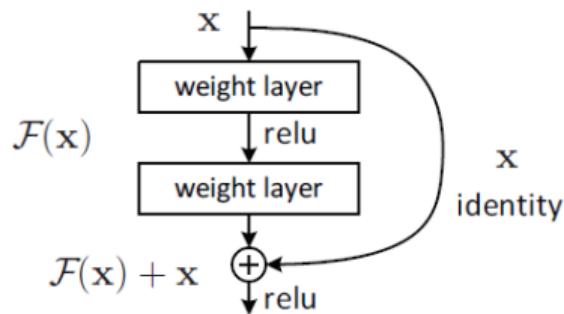


Imagen tomada de He et al. Deep Residual Learning for Image Recognition, 2015

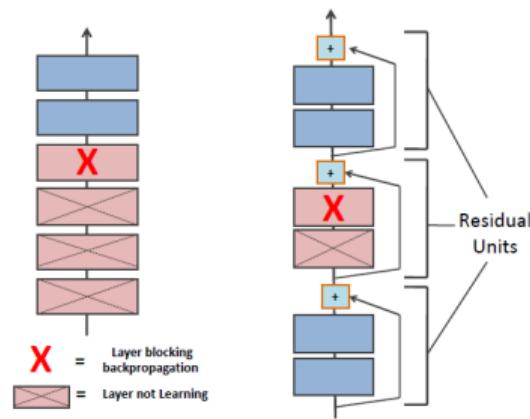


Imagen de Kevin Murphy, tomada de Probabilistic Machine Learning: An Introduction, 2021

Explosión y desvanecimiento del gradiente: inicialización (1)

- ▶ Números aleatorios de distribución gaussiana con media 0 y varianza 0.01.
 - ▶ Funciona en redes pequeñas
 - ▶ Para redes profundas activaciones tienden a volverse 0
- ▶ Números aleatorios de distribución gaussiana con media 0 y varianza 1
 - ▶ Genera saturación de las neuronas y gradientes se vuelven 0

Explosión y desvanecimiento del gradiente: inicialización (2)

- ▶ Para una capa con n_e entradas y n_s salidas
 - ▶ Uniforme de Glorot y Bengio (2010)

$$\theta \sim \mathcal{U} \left[-\sqrt{\frac{6}{n_e + n_s}}, \sqrt{\frac{6}{n_e + n_s}} \right]$$

- ▶ Normal de Glorot y Bengio (2010)

$$\theta \sim \mathcal{N} \left(0, \frac{2}{n_e + n_s} \right)$$

- ▶ Normal de He et al. (2015)

$$\theta \sim \mathcal{N} \left(0, \frac{2}{n_e} \right)$$

Capas de normalización: desplazamiento covariante interno

- ▶ Cambio en distribución de activaciones por cambio en parámetros durante entrenamiento
- ▶ Hace más lento el aprendizaje

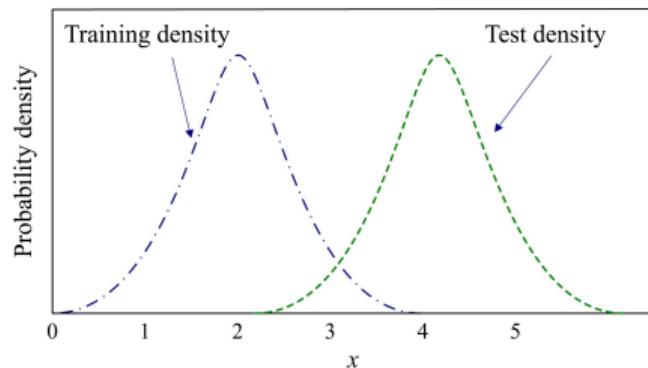


Imagen tomada de Raza et al. EWMA model based shift-detection methods for detecting covariate shifts in non-stationary environments, 2015

Capas de normalización: normalización por lotes (1)

- ▶ Converge más rápido si entradas tienen media 0, varianza 1 y no están correlacionadas
 1. Media del lote

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x^{\{i\}}$$

2. Varianza del lote

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x^{\{i\}} - \mu_{\mathcal{B}})^2$$

3. Normalización

$$\hat{x}^{\{i\}} \leftarrow \frac{x^{\{i\}} - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$

4. Escalado y desplazamiento

$$y^{\{i\}} \leftarrow \gamma \hat{x}^{\{i\}} + \beta$$

Capas de normalización: normalización por lotes (2)

1. Normalizar la red con mini-lote
2. Entrenar la red con retro-propagación
3. Transformar estadísticos del lote a estadísticos de población

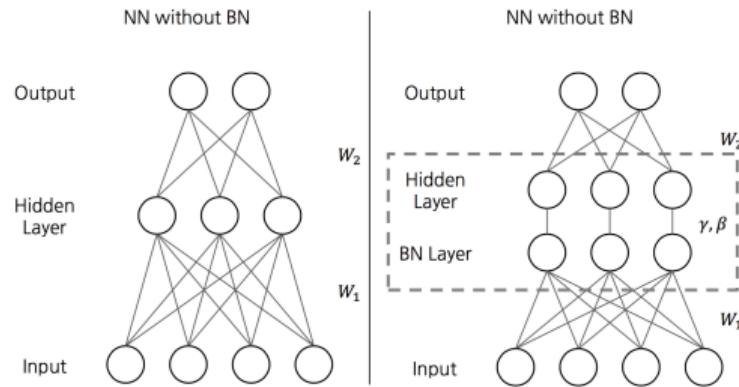


Imagen tomada de <https://wiki.tum.de/display/lfdv/Batch+Normalization>

Capas de normalización: beneficios de la normalización por lotes

- ▶ Acelera el entrenamiento
- ▶ Permite tasas de aprendizaje más grandes
- ▶ Facilita la heurísticas de inicialización de pesos y sesgos
- ▶ Hace posible usar funciones de activación saturadas (por ej. sigmoide)
- ▶ Actúa como un tipo de regularizador
- ▶ Facilita la creación de redes profundas