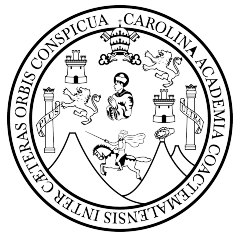
Arquitecturas de Aprendizaje en Redes Neuronales
Curso introductorio

Pedro Elías Mirón Enríquez

pmiron@astro.unam.mx

19 de noviembre de 2024

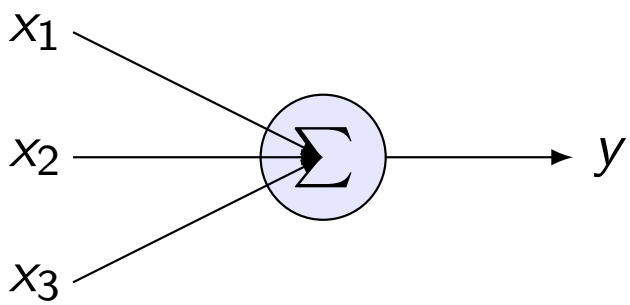


Repaso de ayer

- Énfasis en redes neuronales.
- El ejemplo "más sencillo": el perceptrón.
- Conceptos importantes: entradas, pesos, conexiones, salidas, backpropagation.
- ¿Cómo aprende?

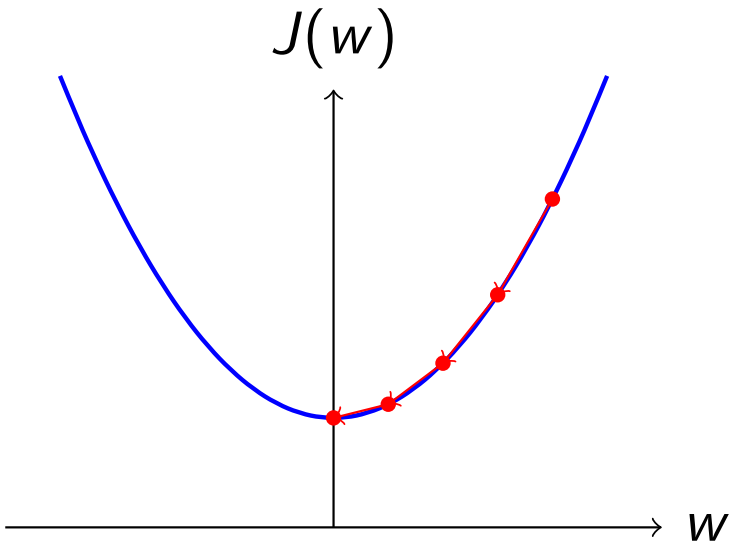
El Perceptrón: El Origen

- Introducido en 1957
- Clasificador binario
- Inspirado biológicamente
- Limitado a problemas linealmente separables



Descenso del Gradiente

- Fundamento del aprendizaje en redes neuronales
- Optimización iterativa para minimizar la función de pérdida
- Ecuación fundamental: $\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$



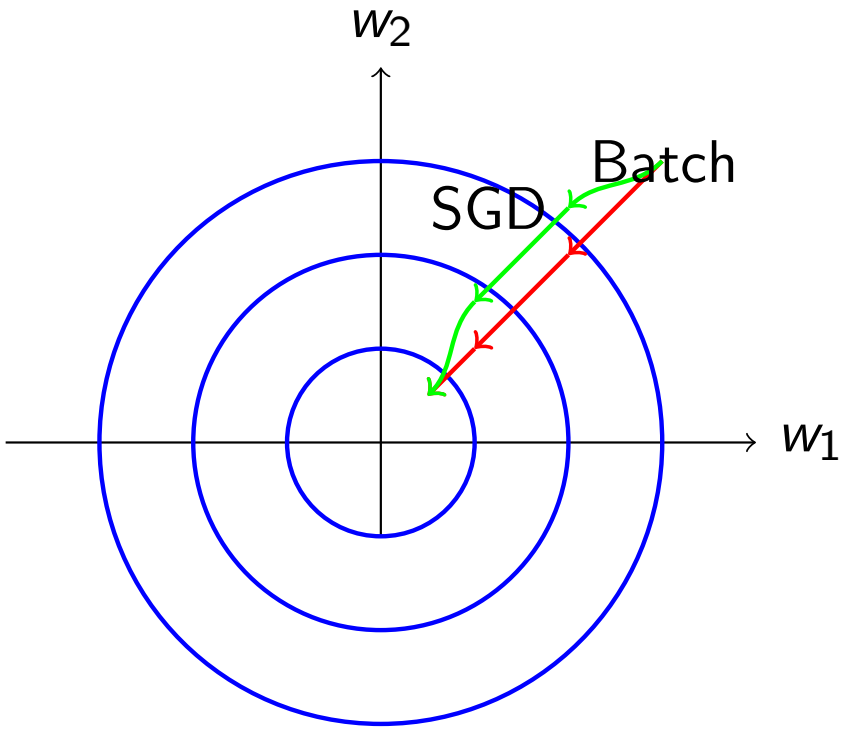
Tipos de Descenso del Gradiente

Por Lotes (Batch)

- Utiliza todo el conjunto de datos
- Más estable
- Mayor costo computacional

Estocástico (SGD)

- Una muestra a la vez
- Más rápido
- Mayor varianza



¿Por qué Backpropagation?

- Algoritmo fundamental para entrenar redes neuronales

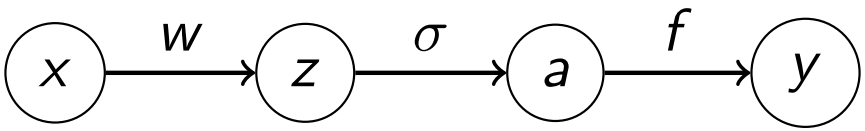
¿Por qué Backpropagation?

- Algoritmo fundamental para entrenar redes neuronales
- Permite ajustar pesos de manera eficiente

¿Por qué Backpropagation?

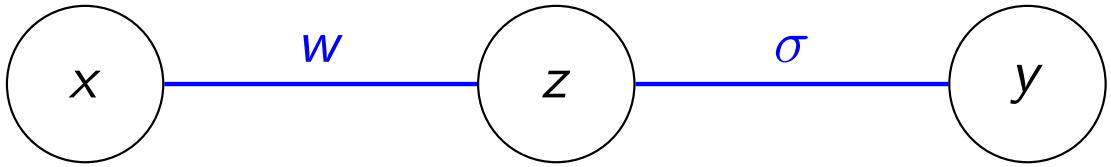
- Algoritmo fundamental para entrenar redes neuronales
- Permite ajustar pesos de manera eficiente
- Basado en la regla de la cadena del cálculo

La Regla de la Cadena



$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w}$$

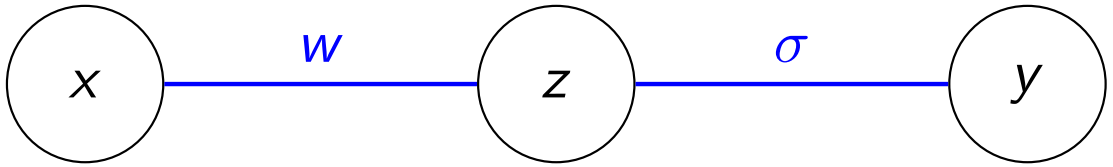
Forward Pass



Entrada Ponderación Salida

1. $z = wx + b$

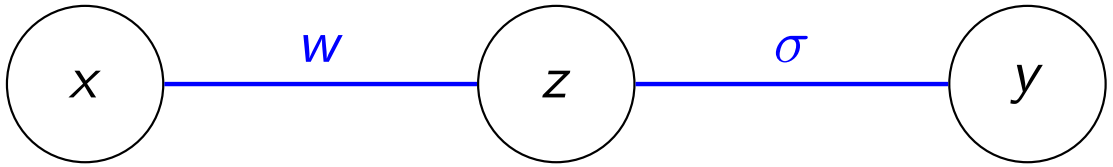
Forward Pass



Entrada Ponderación Salida

1. $z = wx + b$
2. $a = \sigma(z)$

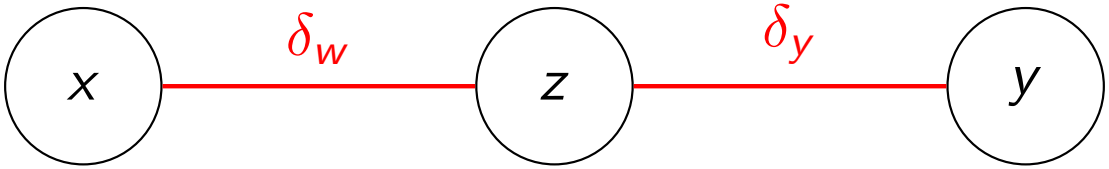
Forward Pass



Entrada Ponderación Salida

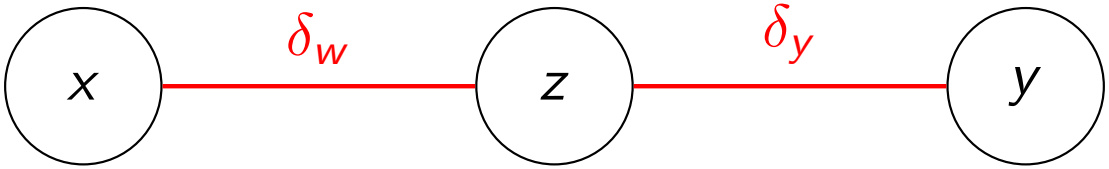
1. $z = wx + b$
2. $a = \sigma(z)$
3. $y = f(a)$

Backward Pass



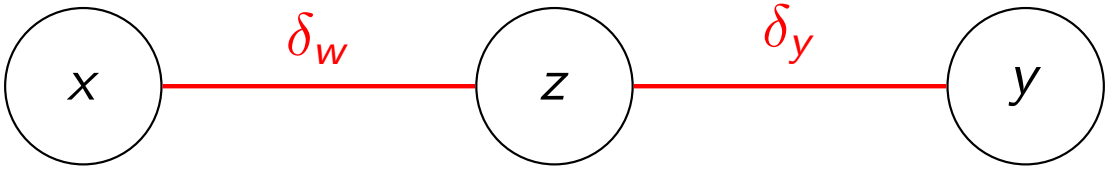
$$\delta_y = -(y_{true} - y)$$

Backward Pass



$$\delta_y = -(y_{true} - y)$$
$$\delta_z = \delta_y \cdot \sigma'(z)$$

Backward Pass



$$\delta_y = -(y_{true} - y)$$
$$\delta_z = \delta_y \cdot \sigma'(z)$$
$$\delta_w = \delta_z \cdot x$$

Funciones de Pérdida Comunes

- Error Cuadrático Medio (MSE):

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Entropía Cruzada:

$$L_{CE} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$$

- Error Absoluto Medio (MAE):

$$L_{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Tasa de Aprendizaje

- Factor crítico en el entrenamiento
- Controla el tamaño de los pasos de actualización
- Impacto en convergencia y estabilidad

Métodos Avanzados de Optimización

- **Adam (Adaptive Moment Estimation)**
  - Combina las ventajas de RMSprop y Momentum
  - Adapta la tasa de aprendizaje para cada parámetro
  - Especialmente útil para datos dispersos
- **AdaGrad**
  - Adapta el learning rate basado en la historia de gradientes
  - Mejor para características infrecuentes

Técnicas de Regularización

- **Dropout**
  - .Apaga.aleatoriamente neuronas durante el entrenamiento
  - Previene el sobreajuste (overfitting)
  - Típicamente se usa p $= 0.5$ para capas ocultas
- **L1 y L2 Regularización**
  - L1: Añade $\lambda \sum |w|$ a la función de costo
  - L2: Añade $\lambda \sum w^2$ a la función de costo
  - Penalizan pesos grandes

Técnicas de Normalización

- **Batch Normalization**
  - Normaliza las activaciones de cada capa
  - Reduce el ïnternal covariate shift"
  - Permite tasas de aprendizaje más altas
- **Layer Normalization**
  - Similar a batch norm pero normaliza a través de las features
  - No depende del tamaño del batch
  - Popular en RNNs y Transformers

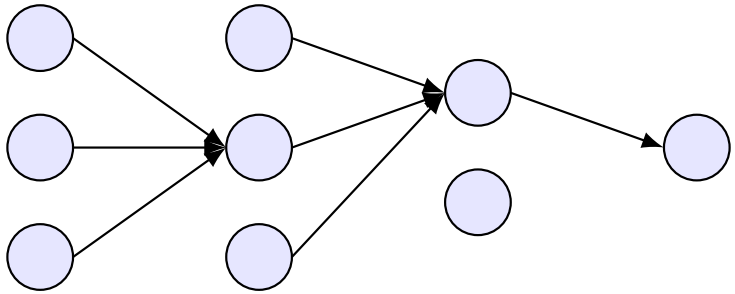
Funciones de Activación Avanzadas

- **LeakyReLU**
  - Soluciona el problema del "dying ReLU"
  - $f(x) = \text{máx}(0{,}01x, x)$
- **ELU (Exponential Linear Unit)**
  - Puede producir salidas negativas
  - Reduce el bias shift
- **SELU (Scaled ELU)**
  - Auto-normalizing neural networks
  - Mantiene media y varianza estables

Redes Neuronales Modernas

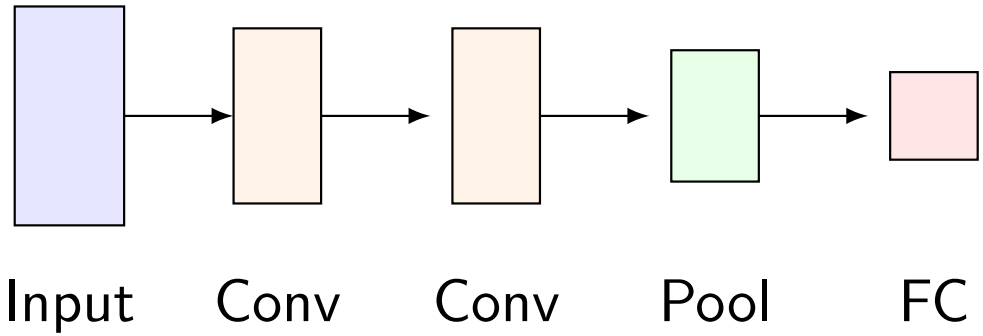
- Transformers
- Redes Generativas Adversarias (GANs)
- Redes Neuronales Gráficas (GNN)
- Arquitecturas de Atención

Redes Multicapa: Rompiendo la Linealidad



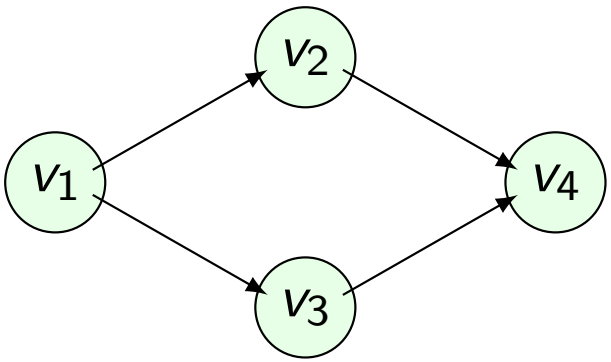
- Múltiples capas de neuronas
- Función de activación no lineal
- Backpropagation
- Universal approximation

CNN: Conquistando la Visión por Computadora



- Convolución: detectores locales de características
- Pooling: reducción y invarianza
- Jerarquía de características

GNN: Redes Neuronales de Grafos



- Procesamiento de grafos
- Agregación de vecinos
- Invarianza permutacional

Referencias

- Goodfellow, I., et al. (2016). Deep Learning
- Vaswani, A., et al. (2017). Attention Is All You Need
- He, K., et al. (2016). Deep Residual Learning
- LeCun, Y., et al. (2015). Deep Learning Nature