

Curso de aprendizaje profundo

PCIC, UNAM

Tarea 1: Redes densas^{*}

1. Red de unidades de umbral lineal

Programa y evalúa una red de neuronas con funciones de activación escalón unitario que aproxime la operación XNOR (\odot) dada por

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	1

Para ello debes asignarle los pesos y sesgos adecuados a cada neurona manualmente. Explica la elección de la red y los valores de los pesos y sesgos.

2. Retropropagación en red densa

Programa el algoritmo de retropropagación usando NumPy para una tarea de clasificación binaria presuponiendo una red densa con dos capas ocultas. Las neuronas de las capas ocultas cuentan con una función de activación ReLU, definida por:

$$ReLU(z) = \max(0, z)$$

Por su parte, la capa de salida está compuesta por una sola neurona logística.

Para el entrenamiento minimiza el promedio de la función de pérdida de entropía cruzada binaria:

$$ECB(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{n} \sum_{i=1}^n \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

Entrena la red mediante descenso por gradiente y el algoritmo de retropropagación de errores. Describe las fórmulas y reglas de actualización de los pesos y sesgos de cada capa y entrena y evalúa la red en algún problema de clasificación no lineal. Compara el comportamiento del entrenamiento de esta red con una en la que las neuronas de las capas ocultas tienen una función de activación logística y en la que la función de pérdida no se promedia.

^{*}Todos los ejercicios tienen el mismo peso

3. Red densa con PyTorch

A partir de la libreta https://github.com/gibranfp/CursoAprendizajeProfundo/blob/2024-1/notebooks/1c_redes_densas_pytorch.ipynb, evalúa distintas configuraciones de redes densas que incluyan lo siguiente:

- Más capas ocultas.
- Distintas funciones de activación de las capas ocultas.
- Técnicas de regularización.
- Capas de normalización.