

Introducción a las Redes Neuronales

Dr. Ariel H. Curiale

Cuatrimestre: 2do de 2018

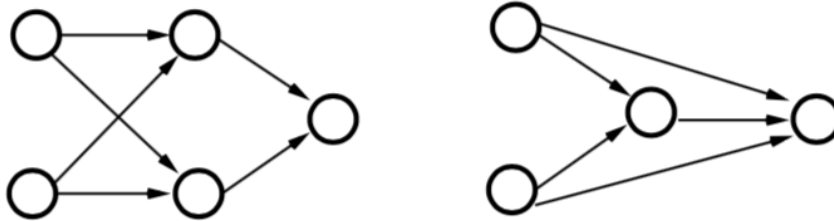


UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO

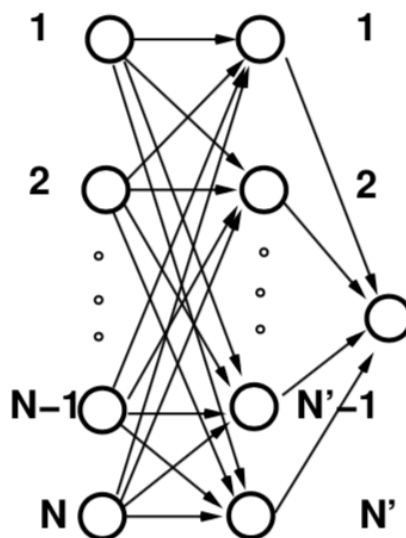


PRÁCTICA: APRENDIZAJE SUPERVISADO

1. La regla XOR tiene dos entradas (1 o -1) y la salida es -1 si ambas son diferentes y 1 si ambas son iguales. Utilizar el algoritmo de retropropagación de errores para aprender el XOR en las siguientes arquitecturas (incluir unidades de entrada adicional para simular los umbrales). Comparar el tiempo de convergencia entre ambas arquitecturas. (Nota: utilizar como función de activación tanh.)

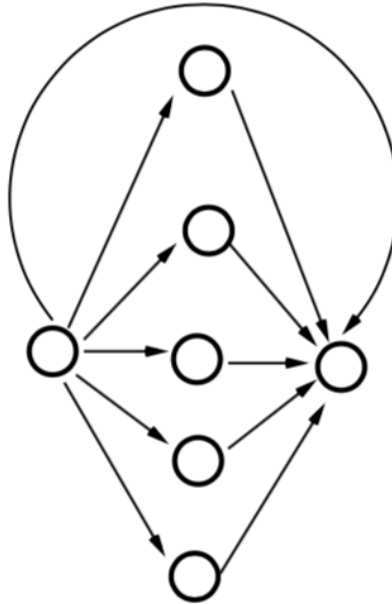


2. El problema de paridad es una generalización del XOR para N entradas. La salida es 1 si el producto de las N entradas es 1 y -1 si el producto de las entradas es -1. Implementar la red neuronal descrita en la gráfica para aprender el problema en la siguiente arquitectura. ¿Que pasa si $N' < N$? (Nota: utilizar como función de activación tanh.)



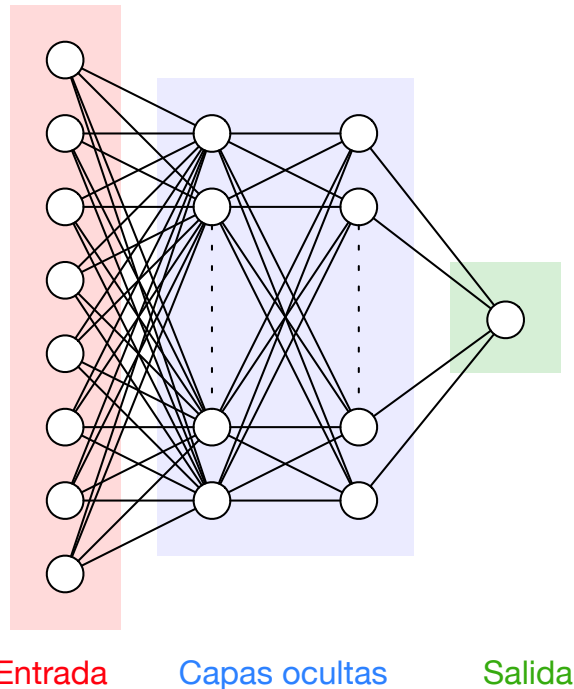
3. Utilizar la arquitectura de la figura para crear una red neuronal que permita aprender el mapeo logístico ($x(t+1) = 4x(t)(1-x(t))$) donde $x \in [0, 1]$ utilizando la siguiente archi-

ectura. La función de activación de la neurona de salida es lineal. Presentar 50 ejemplos y luego testear para ejemplos no presentados. Comparar el error de entrenamiento con el error de generalización. (Nota: utilizar como función de activación tanh en la capa oculta.)



4. Implementar una red neuronal, lo más simple posible, para resolver el problema de una regresión lineal.
5. A partir de los datos registrados en el archivo "pima-indians-diabetes.csv" para cada paciente:
 - Cantidad de embarazos.
 - Concentración de glucosa plasmática a 2 horas en una prueba oral de tolerancia a la glucosa.
 - Presión arterial diastólica (mm Hg).
 - Grosor del pliegue de la piel del tríceps (mm).
 - Insulina en suero de 2 horas (mu U/ml).
 - Índice de masa corporal (peso en kg/(altura en m)²).
 - Función de pedigrí de diabetes (información sobre la historia familiar y la influencia genética para tener diabetes).
 - Edad.

implementar una red neuronal con la siguiente arquitectura (capas ocultas 4 neuronas) para predecir si el paciente puede sufrir de diabetes o no. ¿Que pasa con la precisión si incrementamos el número de neuronas? ¿Y si introducimos una capa oculta más? (Nota: relu como función de activación en las capas ocultas y sigmoid en la salida. Inicializar todas las capas con la clásica dist. uniforme)



6. Estudie el ejemplo que viene incluida en la librería *keras* capaz de reconocer dígitos manuscritos en imágenes (`mnist_cnn_keras.py`) y proponga una arquitectura diferente para resolver el mismo problema. Si quiere puede utilizar la base de datos `mnist` que se encuentra en archivo `datos/mnist_database.npz` o la que viene en `keras`. Grafique la precisión sobre los datos de entrenamiento y los de prueba para cada época de ambas arquitecturas. A partir de los gráficos de rendimiento y el error, analizar pros y cons de las distintas redes neuronales y explique a que se deben las diferencias en el rendimiento. A su vez, explique aara que se utiliza la capa `Dropout` que se encuentra en el ejemplo `mnist_cnn_keras.py` (Nota: utilizar la función `softmax` en la capa de salida.)
7. Proponga una red neuronal capaz de clasificar las imágenes entre perros y gatos. La BD se encuentra en el archivo `cats-dogs_sub.zip`. Utilice la función `imageio.imread` de la librería *imageio* u otra similar para poder leer las imágenes. Recuerde utilizar solamente el 70% de los datos para entrenar, y dejar el 30% para evaluar la precisión de la red propuesta. ¿Cual es el rendimiento si utiliza la arquitectura del punto anterior, explique porque funciona bien o mal?