



IPN-UPIITA

## Redes Neuronales

Reporte R03

Dr. Rafael Martínez Martínez

Academia de sistemas

ramartinezr@ipn.mx

---

### Instrucciones:

- Cada problema/ejercicio debe tener procedimiento ordenado y completo que justifique adecuadamente la respuesta anotada.
  - Si falta el procedimiento o este no justifica la respuesta anotada entonces el problema vale 0 puntos aunque la respuesta sea correcta.
- 

## Contenido

<b>Problema 1 (50 puntos)</b>	1
<b>Problema 2 (50 puntos)</b>	2

### Problema 1 (50 puntos)

En el artículo: [Multilayer feedforward networks are universal approximators](#). Se muestra que las redes feedforward (ya sabemos que es esto ) multicapa, son una clase de aproximadores universales de funciones. A grandes rasgos, esto significa que una red puede aprender a comportarse como cualquier función  $f$ , donde la entrada de la red serían valores del dominio de  $f$ , y la salida de la red sería una aproximación al contradominio de dichos valores (ya entrenada la red).

Entender el artículo podría llevar algo de tiempo, tanto para revisar el background requerido así como entender los argumentos de dicha prueba. En la sección **Function Approximation** (página 11-4) Hagan nos muestra un ejemplo de dicho fenómeno.

Cabe señalar que el ejemplo de Hagan, es muy particular y simplificado, lo cual puede ser suficiente para ilustrar dicho fenómeno. Otros ejemplos más elaborados al respecto se pueden revisar en [A visual proof that neural nets can compute any function](#) elaborado por [Michael Nielsen](#)

Este problema consiste en elaborar una explicación del fenómeno mencionado, apoyado de los enlaces proporcionados (y/u otros). No es necesario incluir cuestiones técnicas del artículo, pero se espera que Hagan sea un punto de referencia y que tomes algunas ideas (gráficas, argumentos, etc.) de Nielsen y si lo consideras necesario de otras referencias (no olvides indicarlo)

## Problema 2 (50 puntos)

Revisa el siguiente video [Universal Approximation-CIS 522 - Deep Learning at the University of Pennsylvania](#) y realiza el ejercicio mencionado al final del video (consiste en aproximar un periodo de la función  $\sin(\pi x)$  mediante funciones ReLU, calcular el error en función del número de segmentos usados para la aproximación y plotarlo usando MATLAB/OCTAVE/PYTHON/OTRO).