

# MEJORAS EN LAS NUEVAS TÉCNICAS DE INTELIGENCIA ARTIFICIAL PARA LA DETECCIÓN DE ANOMALÍAS EN REACTORES NUCLEARES (INAIA)

**Gumersindo J. Verdú Martín**

gverdu@iqn.upv.es

---

**Instituto Universitario de Seguridad Industrial, Radiofísica y Medioambiental (ISIRYM)  
Universitat Politècnica de València**



Jornada anual de I+D+i

Febrero 2023



# Índice

- Motivación
- Objetivos
- Librerías empleadas
- Datos simulados: Caso 2D
- Diseño de la Red Neuronal
- Evaluación y Resultados
- Conclusiones
- Trabajo en Desarrollo

# Motivación

- El análisis del ruido de neutrónico estudia los efectos de pequeñas perturbaciones en los reactores nucleares.
- Estas perturbaciones generalmente se expresan como cambios en las secciones eficaces macroscópicas y pueden ser causadas por fluctuaciones estocásticas de las propiedades del refrigerante y/o la vibración mecánica de los elementos combustibles.
- Análisis del ruido de neutrónico: **monitoreo no invasivo, detección y ubicación de anomalías en reactores nucleares.**

# Objetivos

- **Desarrollo de herramientas basadas en técnicas de Deep Learning para el estudio y monitorización del estado del núcleo de reactores nucleares.**
- Detectar el lugar donde se ha producido una perturbación.
- Detectar qué tipo de perturbación es.
- Estudiar el nivel máximo de ruido en la señal que se puede llegar a clasificar correctamente.
- Estudiar cómo afecta la disminución de detectores activos en la clasificación.

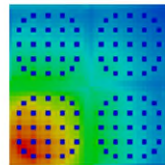
# Librerías empleadas

- Librería de Deep Learning **Keras**: sobre librerías Theano o Tensorflow.
- Modularidad, minimalismo, extensibilidad, y lenguaje **Python**.
- Soporte CPU y GPU.
- Entrenamiento y clasificación llevado a cabo en **GPU NVIDIA A100 TESLA EDU** PCI-E 4.0 con 40GB de memoria, 6.912 CUDA cores y 640 Tensor cores.



# Librerías empleadas

- Generación de datos: **FEMFFUSION**.
- Código de elementos finitos para el modelado de sistemas nucleares.
- Escrito en lenguaje de programación C++.
- Librería Open Source.
- Desarrollada en la Universitat Politècnica de València.



## FEMFFUSION

a finite element method code for nuclear reactor modelling

<https://femffusion.webs.upv.es/>

# Datos simulados: Caso 2D

- Reactor BIBLIS 2D : 257 celdas con hasta 257 detectores activos
- Dos tipos de perturbaciones introducidas:

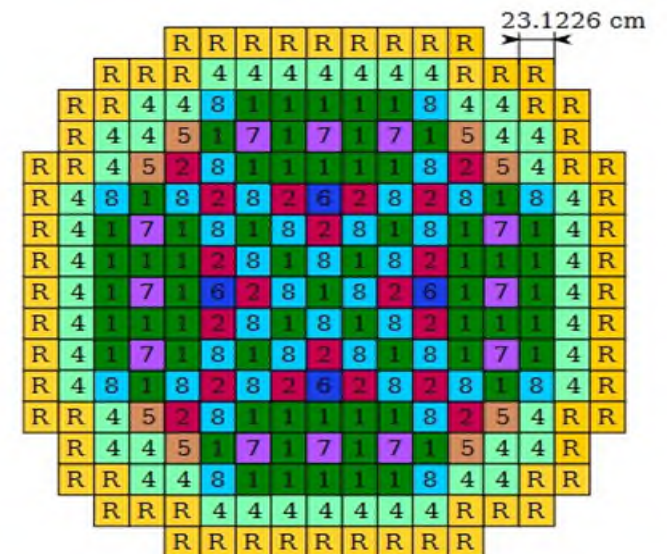
Cambio de la sección eficaz de dispersión  $\Sigma_a(t) = \Sigma_a0 + 0.01\Sigma_a0 \sin(2\pi ft)$  en la celda (i, j)

Cambio de la sección eficaz de absorción  $\Sigma_s(t) = \Sigma_s0 + 0.01\Sigma_s0 \sin(2\pi ft)$  en la celda (i, j)

- Diferentes frecuencias:  $f = 0.1, 0.5, 1, 5$  y  $10$  Hz
- Diferentes ubicaciones de la perturbación: 257 posibles
- Por cada perturbación se obtienen los siguientes datos:

1. Neutron Power
2. Group 1 flux ( $\phi_1$ )
3. Group 2 flux ( $\phi_2$ )
4. Flux Noise Group 1 ( $\delta\phi_1$ )
5. Flux Noise Group 2 ( $\delta\phi_2$ )

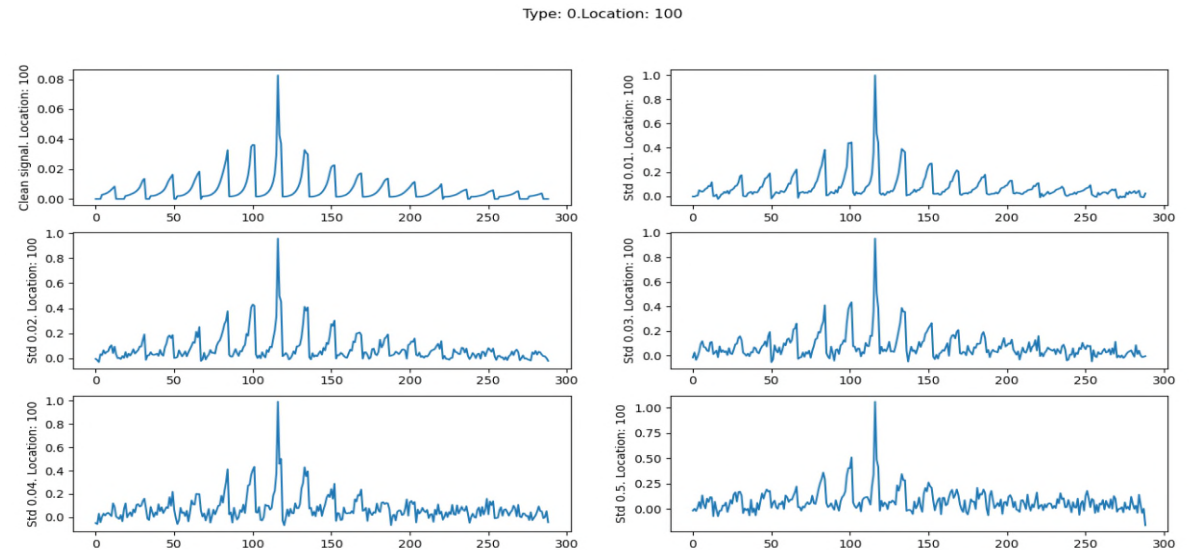
- **TOTAL: 2570 señales**



Reactor BIBLIS 2D por materiales.

# Datos simulados: Caso 2D

- Adición del ruido Gaussiano:
  - ✓ Se añade ruido a las señales para no tener solo casos ideales
  - ✓ Se amplía el conjunto de señales para tener más datos
- Desviación estándar de 0.01 a 0.05 en pasos de 0.005
- **TOTAL: 25700 señales**

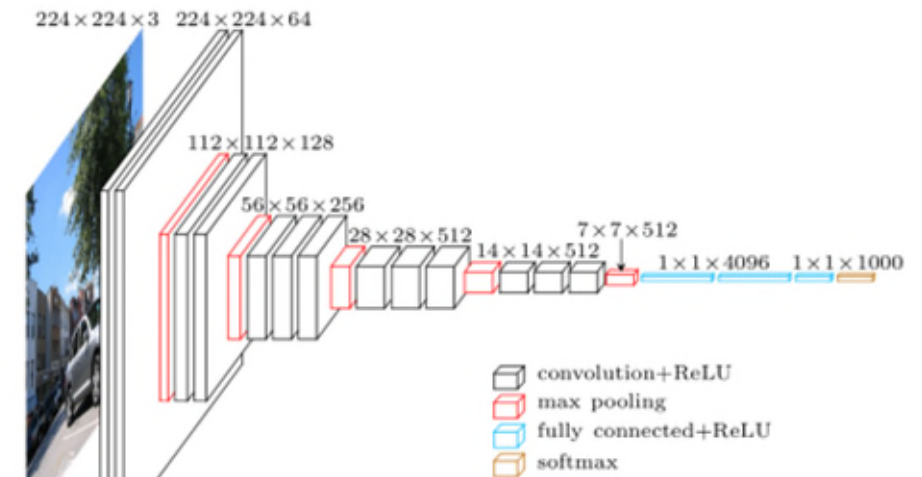


Adición de ruido a las señales.



# Diseño de la red neuronal

- Prueba: Red Neuronal Convolutiva VGG16
  - Red de clasificación de imágenes RGB en 1000 clases
  - 16 capas (convolucionales y densas) combinadas con capas de *Max Pooling* (submuestreo)
  - Última capa *Softmax* para clasificar según probabilidades
  - Precisión del 92.7% con *dataset ImageNet* (14 millones de imágenes)



Red VGG16.

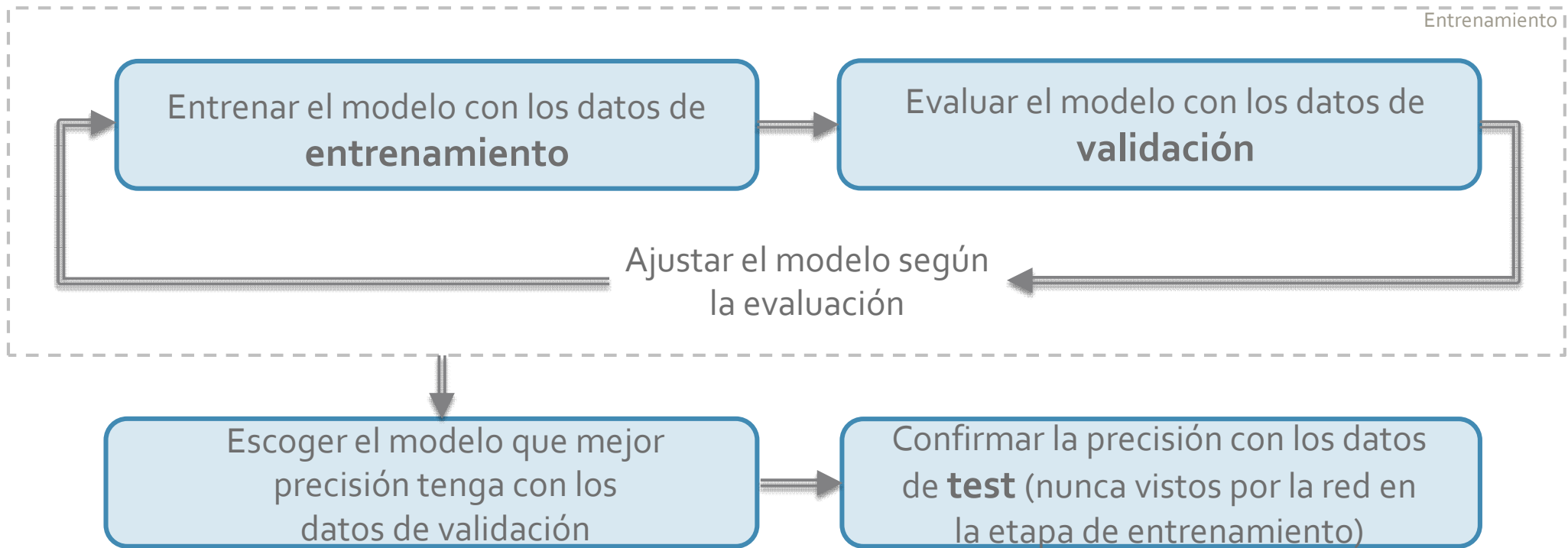
Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014) .

# Diseño de la red neuronal

- Datos de entrada:
  - $\frac{|\delta\phi_2|}{\phi_2}$  para cada perturbación
  - Señales unidimensionales 1x257 → Matriz bidimensional 17x17
  - Celdas sin detectores → valores a 0
  - Se conserva la información espacial
  - Normalización de las señales entre 0 y 1
- División en tres conjuntos:
  - Datos de entrenamiento: 67.5%
  - Datos de validación: 22.5%
  - Datos de test: 10%

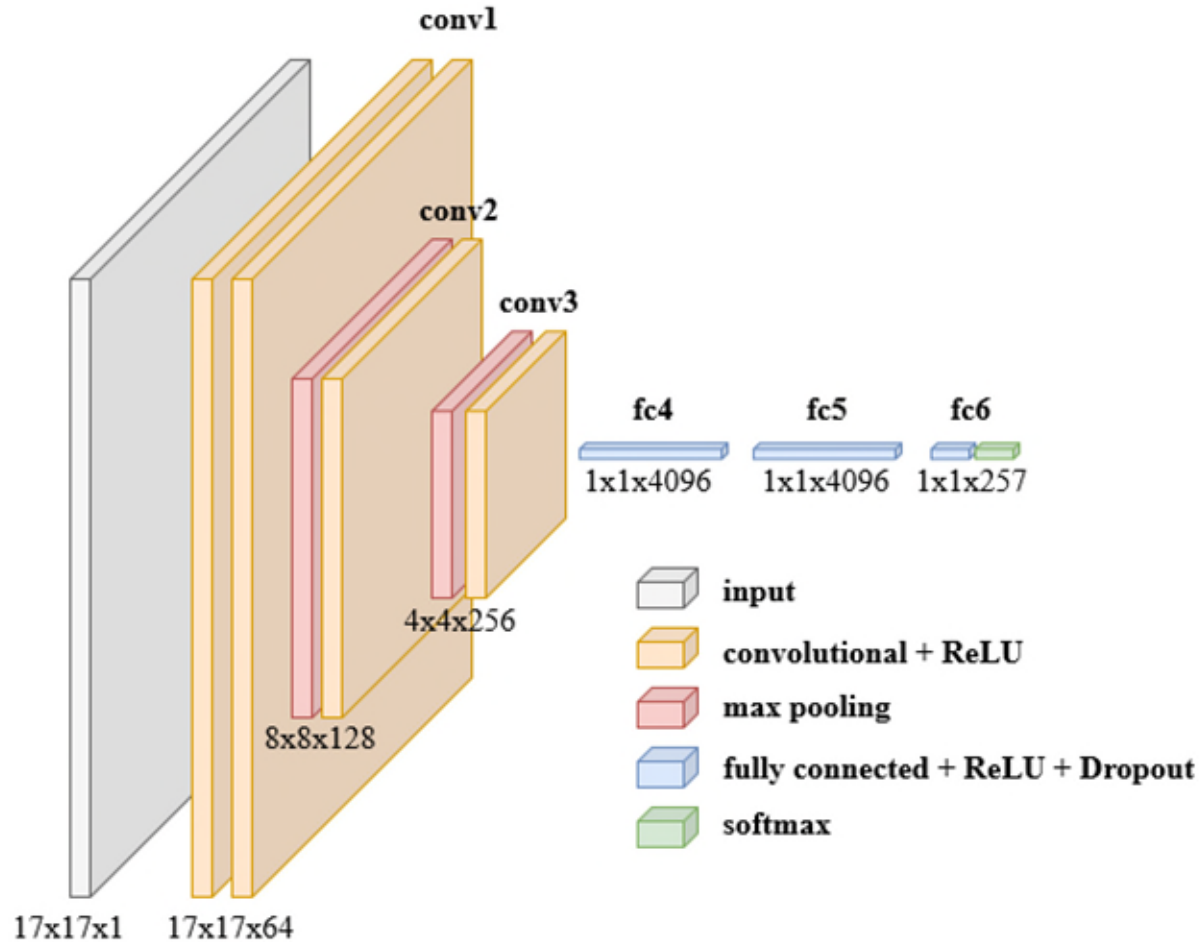
# Diseño de la red neuronal

- Importancia de la división de los datos en conjuntos o *sets*:



# Diseño de la red neuronal

- Detección **lugar** de perturbación



- ✓ Señales etiquetadas por lugar de origen de la perturbación
- ✓ Balanceo entre los grupos (aprox. mismo número de señales de cada origen en cada grupo)
- ✓ Fases de *Dropout* para reducir el *overfitting*
- ✓ Última capa activación *Softmax* para 257 clases

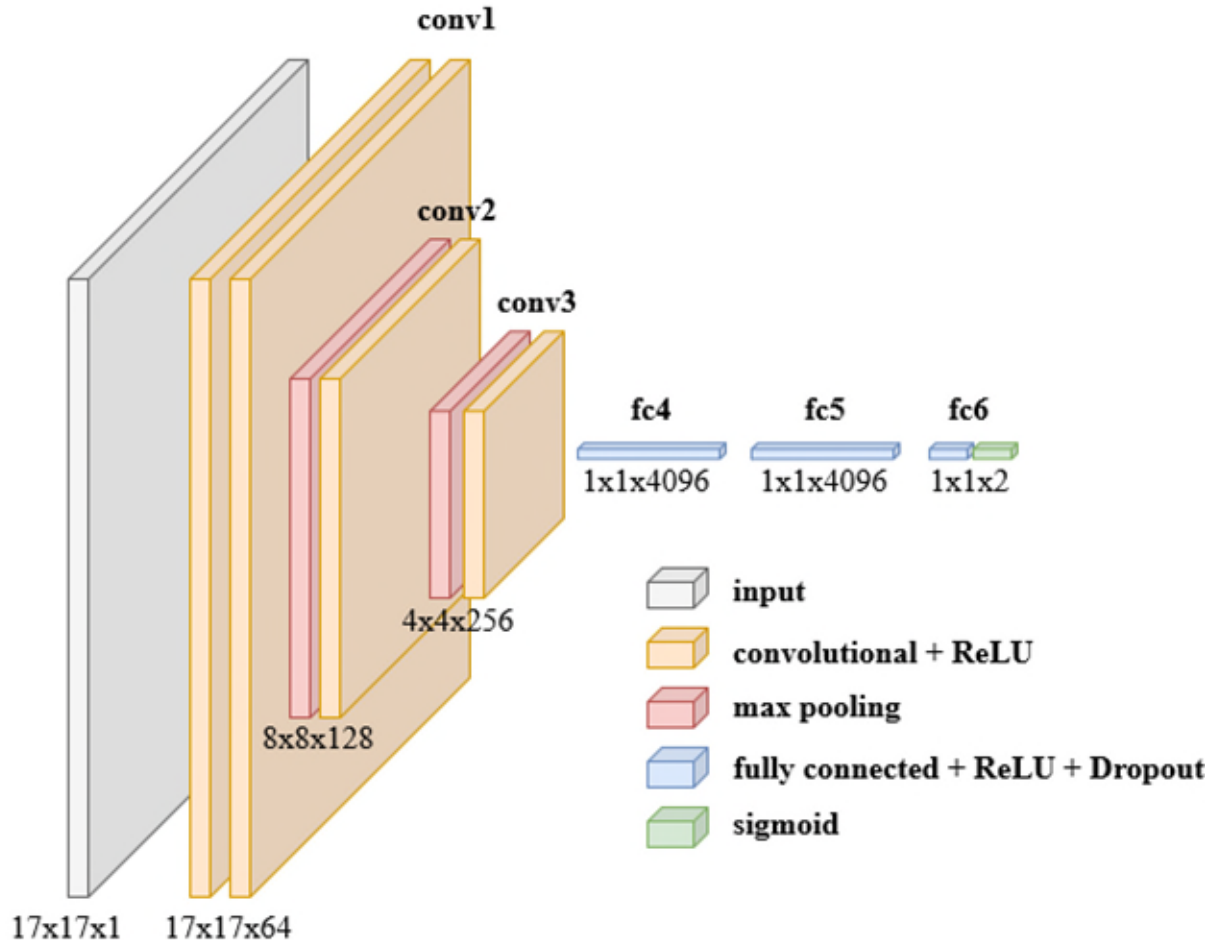
# Diseño de la red neuronal

- Detección **lugar** de perturbación

```
Layer (type)                Output Shape                Param #  
-----  
conv2d (Conv2D)             (None, 17, 17, 64)         640  
conv2d_1 (Conv2D)           (None, 17, 17, 64)         36928  
max_pooling2d (MaxPooling2D) (None, 8, 8, 64)           0  
conv2d_2 (Conv2D)           (None, 8, 8, 128)          73856  
max_pooling2d_1 (MaxPooling2D) (None, 4, 4, 128)          0  
conv2d_3 (Conv2D)           (None, 4, 4, 256)          295168  
flatten (Flatten)           (None, 4096)                0  
dense (Dense)                (None, 4096)                16781312  
dropout (Dropout)           (None, 4096)                0  
dense_1 (Dense)              (None, 4096)                16781312  
dropout_1 (Dropout)         (None, 4096)                0  
dense_2 (Dense)              (None, 257)                 1052929  
-----  
Total params: 35,022,145  
Trainable params: 35,022,145  
Non-trainable params: 0
```

# Diseño de la red neuronal

- Detección de **tipo** de perturbación



- ✓ Señales etiquetadas por tipo de la perturbación
- ✓ Balanceo entre los grupos (aprox. mismo número de señales de cada tipo en cada grupo)
- ✓ Fases de *Dropout* para reducir el *overfitting*
- ✓ Última capa activación sigmoide para 2 clases

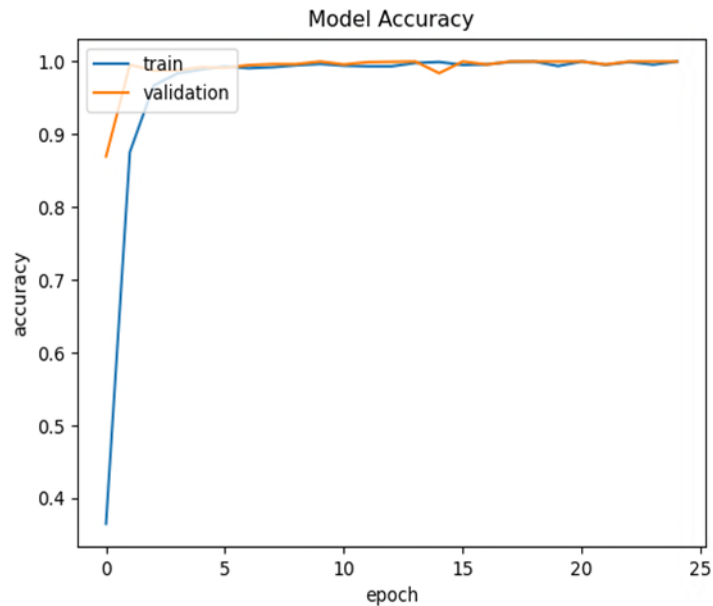
# Diseño de la red neuronal

- Detección de **tipo** de perturbación

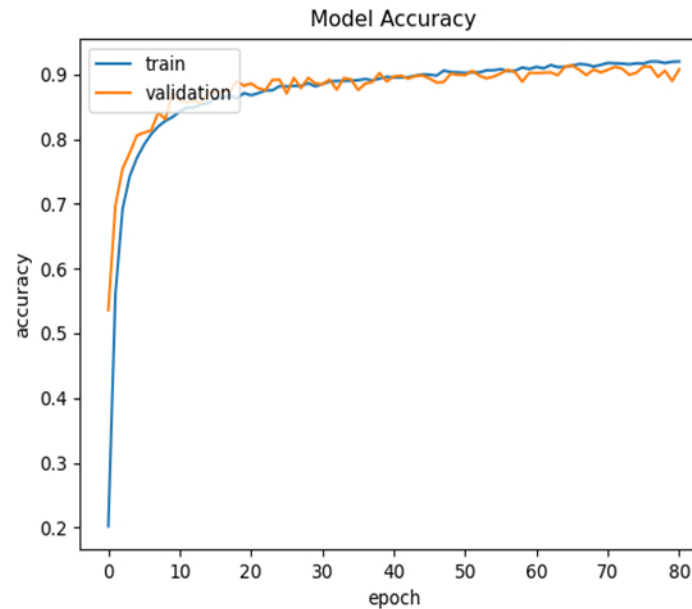
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 17, 17, 64)	640
conv2d_1 (Conv2D)	(None, 17, 17, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_2 (Conv2D)	(None, 8, 8, 128)	73856
max_pooling2d_1 (MaxPooling2D)	(None, 4, 4, 128)	0
conv2d_3 (Conv2D)	(None, 4, 4, 256)	295168
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 4096)	16781312
dropout (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16781312
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 2)	8194
-----		
Total params: 33,977,410		
Trainable params: 33,977,410		
Non-trainable params: 0		

# Evaluación y Resultados

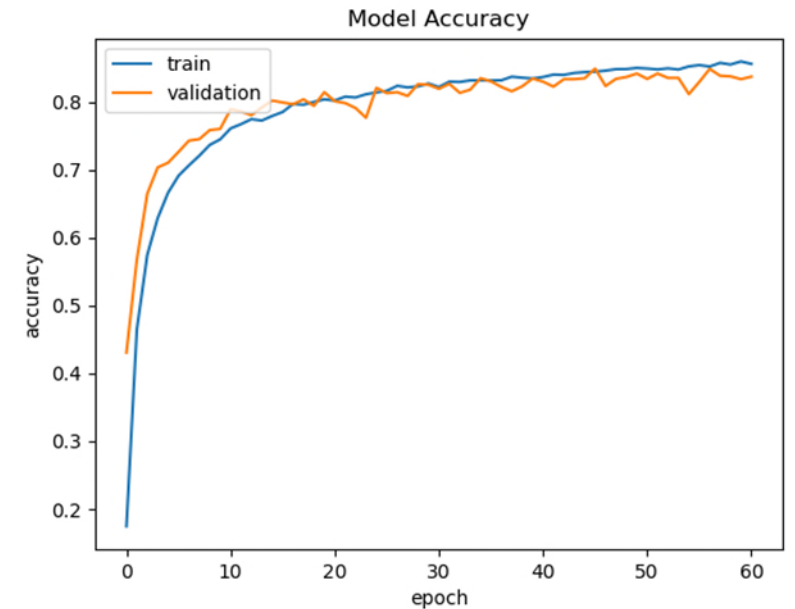
- Entrenamiento detección **lugar** de perturbación



Precisión por etapas de entrenamiento. Todos los detectores activos. Señales con ruido.



Precisión por etapas de entrenamiento. Detectores tipo 1 activos. Señales con ruido.



Precisión por etapas de entrenamiento. Detectores tipo 5 y 7 activos. Señales con ruido.



# Evaluación y Resultados

- Resultados detección **lugar** de perturbación

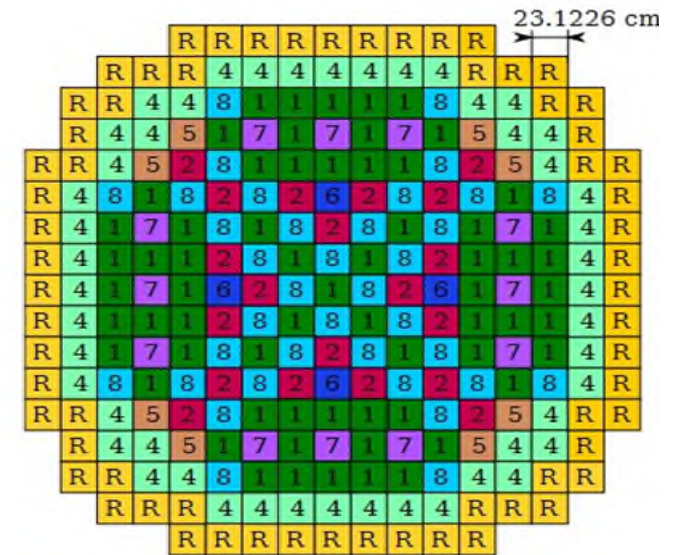
*Distancia Euc.* =  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ , siendo  $(x_2, y_2)$  las coordenadas de la predicción y  $(x_1, y_1)$  las del lugar correcto.

Material	Detectores Activos	Precisión (Test set)	Distancia Media (Error)
<b>Todos</b>	257	1	0
1	65	0.999	0.0005
4	48	1	0
8	36	0.994	0.006
2	20	0.959	0.05
5,7	20	0.995	0.006
7	12	0.961	0.05
5,6	12	0.957	0.05
5	8	0.959	0.04
6	4	0.920	0.14

Tabla 1: Precisión localización disminuyendo el número de detectores activos (señales sin ruido).

Material	Detectores Activos	Precisión (Test set)	Distancia Media (Error)
<b>Todos</b>	257	1	0
1	65	0.911	0.12
4	48	0.966	0.04
8	36	0.820	0.25
2	20	0.715	0.41
5,7	20	0.842	0.20
7	12	0.732	0.43
5,6	12	0.720	0.38
5	8	0.616	0.55
6	4	0.370	1.39

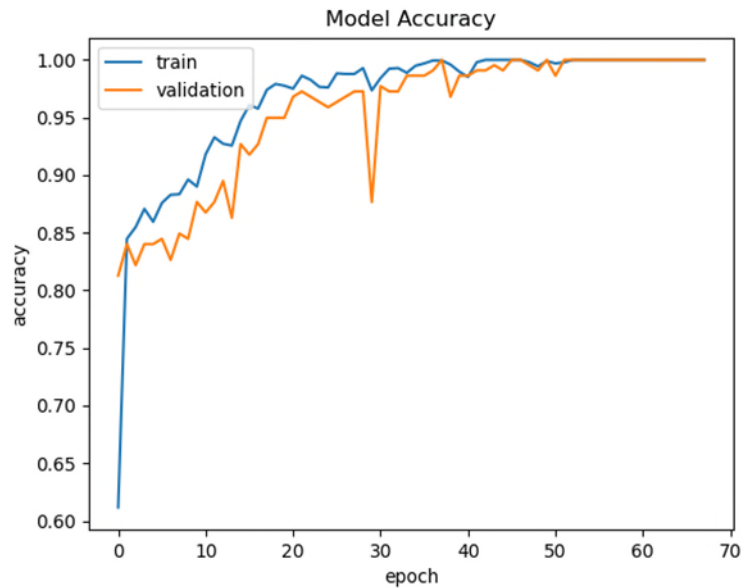
Tabla 2: Precisión localización disminuyendo el número de detectores activos (señales con ruido).



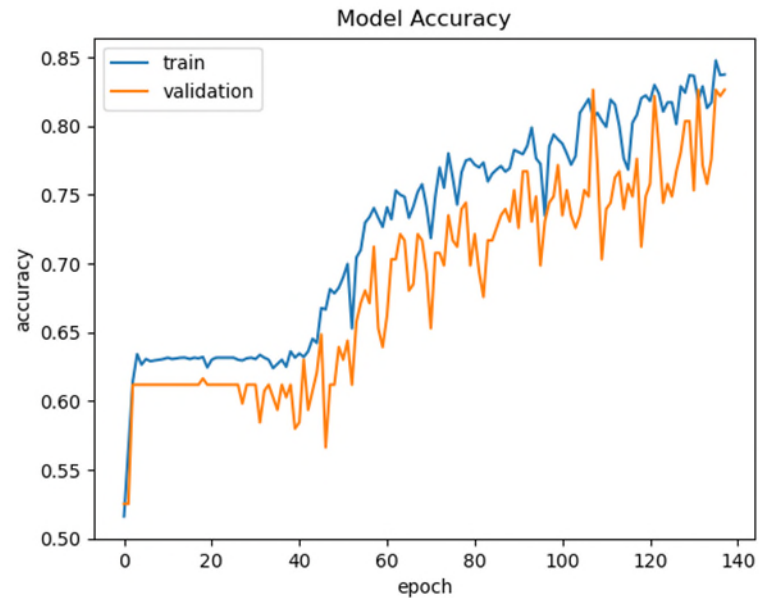
Reactor BIBLIS 2D por materiales.

# Evaluación y Resultados

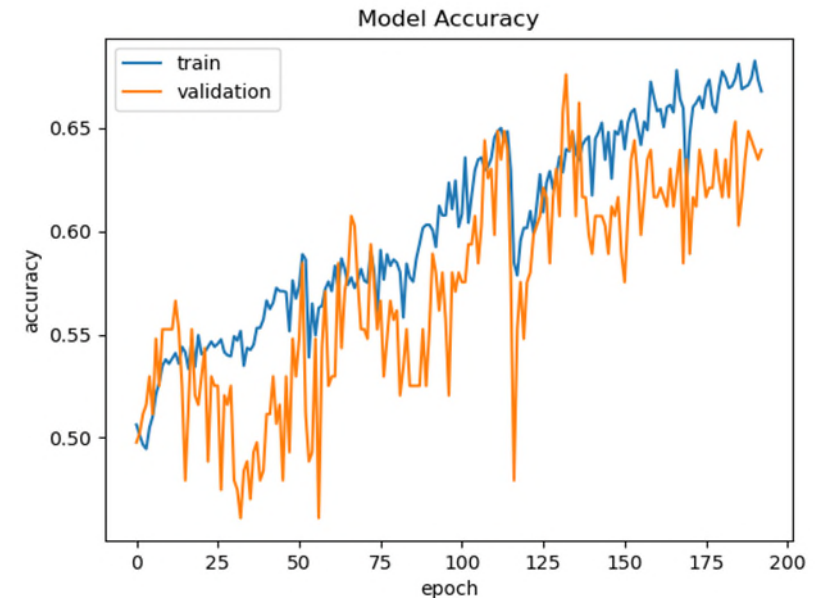
- Entrenamiento detección **tipo** de perturbación



Precisión por etapas de entrenamiento.  
Todos los detectores activos. Señales con ruido.



Precisión por etapas de entrenamiento.  
Detectores tipo 1 activos. Señales con ruido.



Precisión por etapas de entrenamiento.  
Detectores tipo 5 y 7 activos. Señales con ruido.

# Evaluación y Resultados

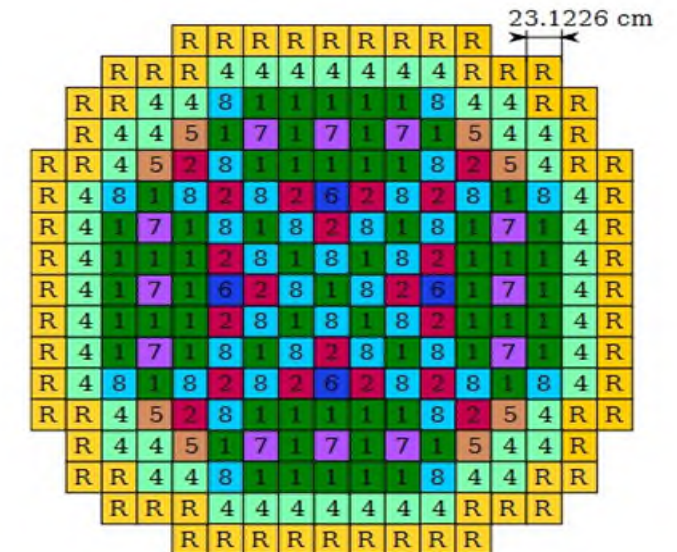
- Resultados detección **tipo** de perturbación

Material	Detectores Activos	Precisión (Test set)
<b>Todos</b>	257	1
1	65	0.938
8	36	0.810
2	20	0.660
5,7	20	0.853
7	12	0.552
5,6	12	0.656
5	8	0.601
6	4	0.506

Tabla 3: Precisión tipo disminuyendo el número de detectores activos (señales sin ruido).

Material	Detectores Activos	Precisión (Test set)
<b>Todos</b>	257	1
1	65	0.784
8	36	0.700
2	20	0.608
5,7	20	0.590
7	12	0.547
5,6	12	0.536
5	8	0.500
6	4	0.500

Tabla 4: Precisión tipo disminuyendo el número de detectores activos (señales con ruido).



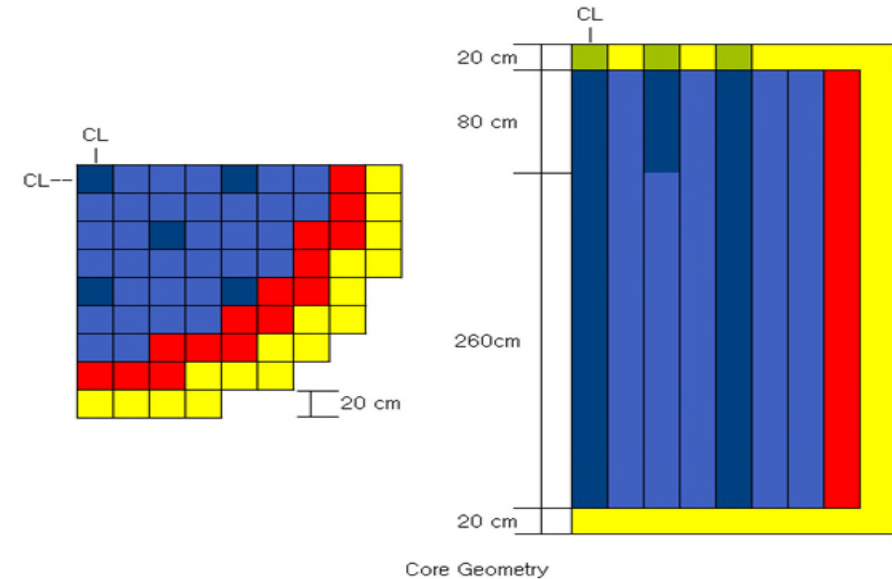
Reactor BIBLIS 2D por materiales.

# Conclusiones

- Primera aproximación al uso de Redes Neuronales Convolucionales para la detección y caracterización de perturbaciones.
- Importante: generación, análisis y preparación de los datos necesarios para el entrenamiento y clasificación.
- Caso en 2D del reactor BIBLIS resuelto con CNN de 2 dimensiones.
- Baja precisión al disminuir el número de detectores activos para clasificar por tipo.
- Sin embargo, errores pequeños en la localización, sobretodo en señales sin ruido.
- Punto de partida para los casos 3D.

# Trabajo en Desarrollo

- Análisis y Optimización de las redes 2D empleadas.
- Generación de los datos para un caso 3D:
  - **Reactor 3D IAEA PWR.**
  - Tiene 19 planos de 17x17 celdas, 4579 posibles posiciones de las perturbación.
  - Tres tipos de perturbaciones. Se añaden las axiales.
  - Total 52000 señales generadas.
- Adaptación de las CNNs para este caso 3D.
- Estudio de nuevas técnicas de Deep Learning.



Two Group Cross Sections for Each Composition

Material	$D_0$	$\Sigma_{a0}$	$\nu\Sigma_{f0}$	$\Sigma_{a2<-1}$
Fuel1	1,500	0,010	0,000	0,020
	0,400	0,085	0,135	
Fuel1+Rod	1,500	0,010	0,000	0,020
	0,400	0,130	0,135	
Fuel2	1,500	0,010	0,000	0,020
	0,400	0,080	0,135	
Reflector	2,000	0,000	0,000	0,040
	0,300	0,010	0,000	
Reflector+Rod	2,000	0,000	0,000	0,040
	0,300	0,055	0,000	

Reactor 3D IAEA PWR.

# Gracias por su atención

Gumersindo J. Verdú Martín  
gverdu@iqn.upv.es



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA