

## TP4

Le TP est dédié à la construction de CNNs pour l'approximation de fonctions telles que  $x \mapsto x^3$ .

Un point technique important est d'utiliser une construction emboîtée des réseaux, comme dans l'exemple ci-dessous en passant par la déclaration **layers** (qui remplace **Sequential**) et la fonction **concatenate** qui permet de composer des réseaux entre eux. La déclaration **layers** permet d'enchaîner les couches sans difficulté.

---

```
from tensorflow.keras import layers

#### x-> x^2 ####
model2=layers.Dense(3, activation='relu',
                    kernel_initializer=init_W0,
                    use_bias=True, bias_initializer=init_b0)(inputs)

model=layers.Dense(1, activation='relu',
                  kernel_initializer=init_W_sortie,
                  use_bias=False)(model2)

## x->x ###
initializer = tensorflow.keras.initializers.Constant(1.)
identity=layers.Dense(1, activation='linear',
                     kernel_initializer=initializer,
                     use_bias=False)(inputs)

### x-> (x-x^2,x)
Lign1=layers.concatenate([model, identity])
```

---

- Charger <https://github.com/despresbr/NNNA/blob/main/mul.py> et exécuter pour afficher la fonction  $f$ .
- Analyser en détail la structure des instructions.
- Augmenter le nombre de couches dans  $x \mapsto x^2$ , et afficher l'erreur finale en fonction du nombre de couches.
- Au choix, réaliser une approximation de  $x \mapsto x^5$  ou réaliser un multiplieur  $(x, y) \mapsto xy$ .