

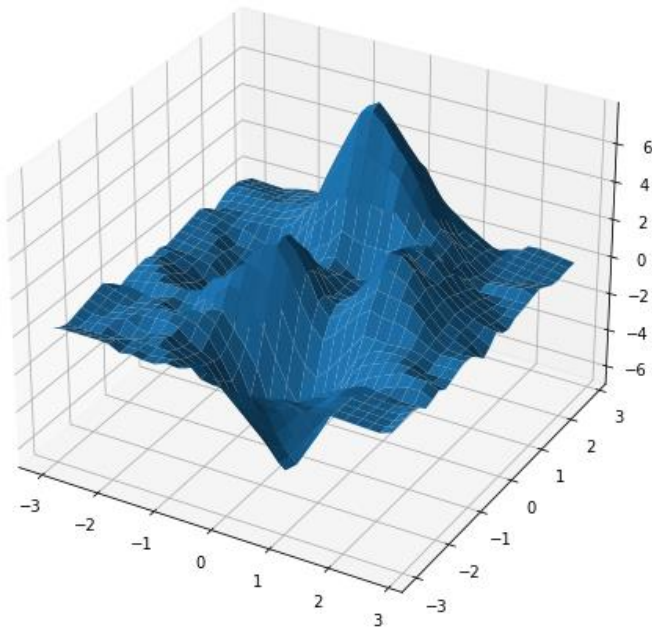
Explique detenidamente como llegó a la mejor solución, tal como las diferentes arquitecturas probadas, épocas de entrenamiento, gráficas obtenidas hasta el mejor resultado. Escriba un documento explicando paso a paso su experimentación y los resultados que vaya obteniendo

- **Realizando pruebas con diferentes arquitecturas:**

- Con una arquitectura de una capa oculta de 50 neuronas:

```
# Se crea la red neuronal de regresión
net = MLPRegressor(solver = 'lbfgs', max_iter=10000, alpha = 1e-3, hidden_layer_sizes = (50,))
```

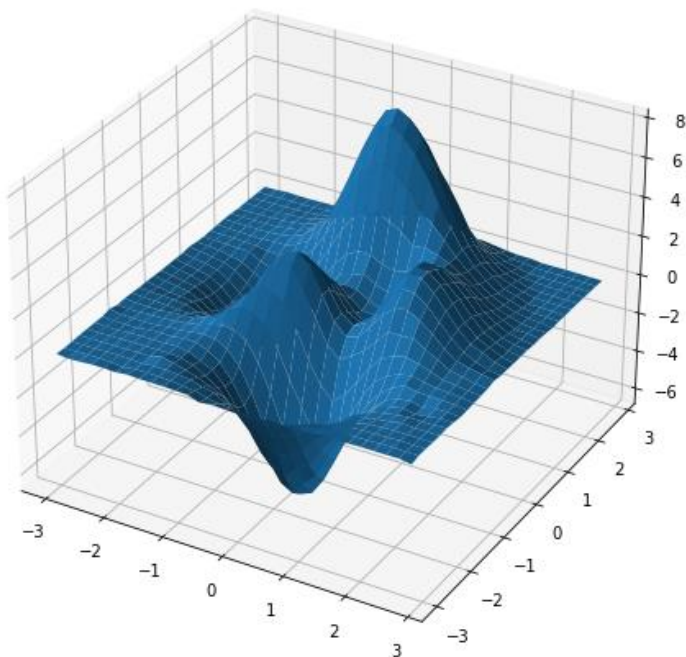
Error de generalización: 0.08406644139655424



- Con una arquitectura de dos capas ocultas de 50 neuronas cada una:

```
# Se crea la red neuronal de regresión
net = MLPRegressor(solver = 'lbfgs', max_iter=10000, alpha = 1e-3, hidden_layer_sizes = (50,50))
```

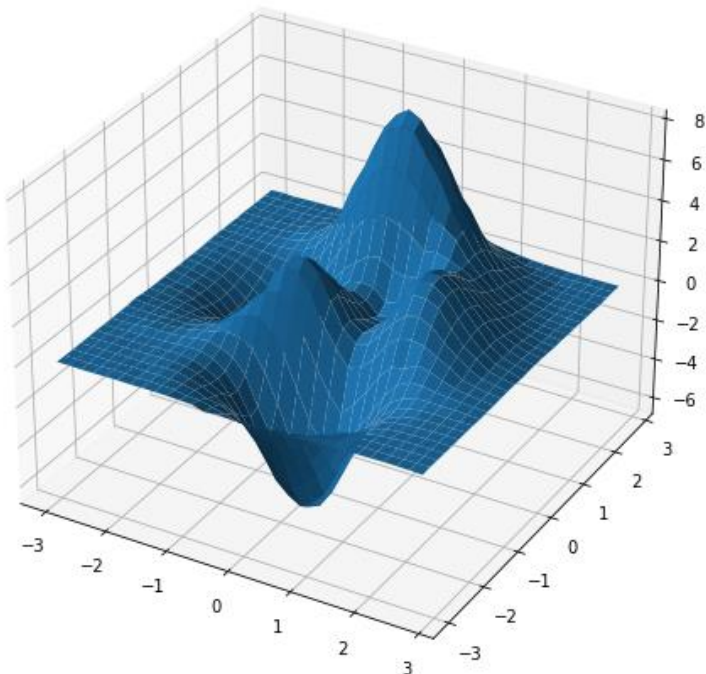
Error de generalización: 0.013577606821198634



- Con una arquitectura de tres capas ocultas de 50 neuronas cada una:

```
# Se crea la red neuronal de regresión
net = MLPRegressor(solver = 'lbfgs', max_iter=10000, alpha = 1e-3, hidden_layer_sizes = (50,50,50))
```

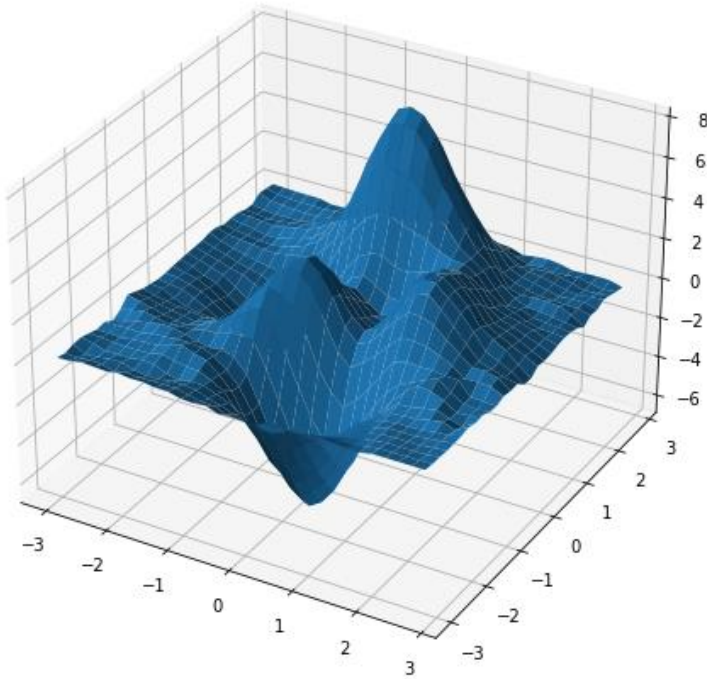
Error de generalización: 0.008260339681019562



- Con una arquitectura de una capa oculta de 100 neuronas:

```
# Se crea la red neuronal de regresión
net = MLPRegressor(solver = 'lbfgs', max_iter=10000, alpha = 1e-3, hidden_layer_sizes = (100,))
```

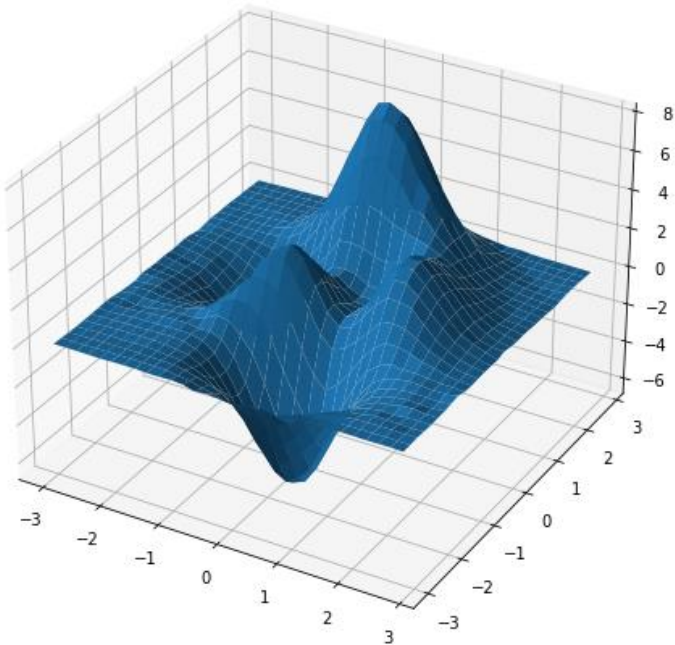
Error de generalización: 0.04015233320684859



- Con una arquitectura de una capa oculta de 100 neuronas y una capa oculta de 50 neuronas:

```
# Se crea la red neuronal de regresión  
net = MLPRegressor(solver = 'lbfgs', max_iter=10000, alpha = 1e-3, hidden_layer_sizes = (100,50))
```

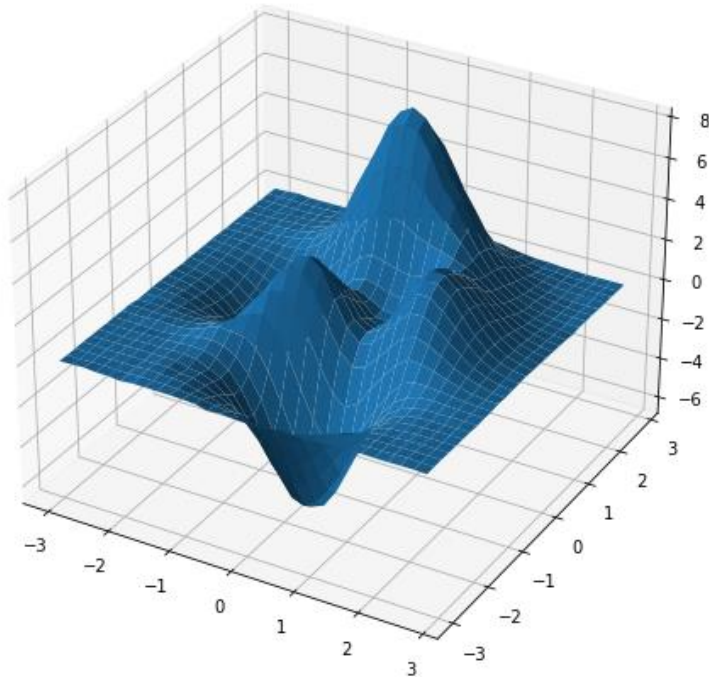
Error de generalización: 0.008572439181630262



- Con una arquitectura de una capa oculta de 100 neuronas y dos capas ocultas de 50:

```
# Se crea la red neuronal de regresión  
net = MLPRegressor(solver = 'lbfgs', max_iter=10000, alpha = 1e-3, hidden_layer_sizes = (100,50,100))
```

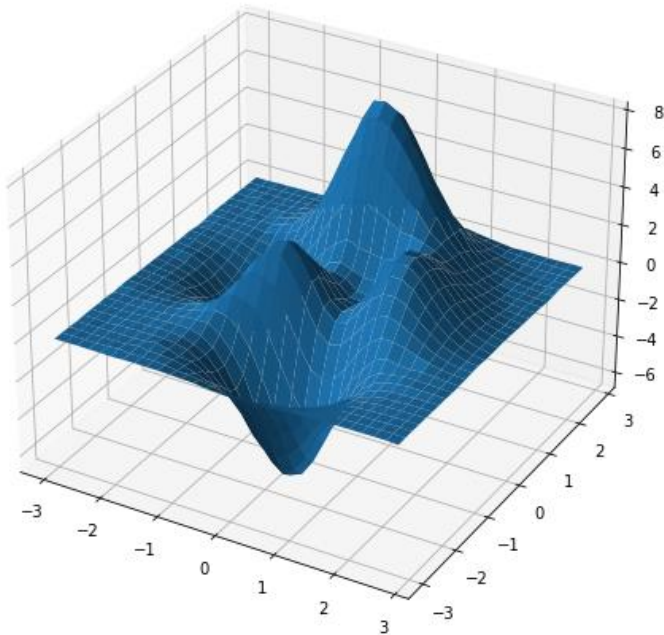
Error de generalización: 0.0062653527629467475



- Con una arquitectura de tres capas ocultas de 100 neuronas cada una:

```
# Se crea la red neuronal de regresión  
net = MLPRegressor(solver = 'lbfgs', max_iter=10000, alpha = 1e-3, hidden_layer_sizes = (100,100,100))
```

Error de generalización: 0.005642310768935724



La arquitectura con tres capas ocultas de 100 neuronas cada una obtuvo el menor error de generalización, lo que indica un mejor rendimiento.

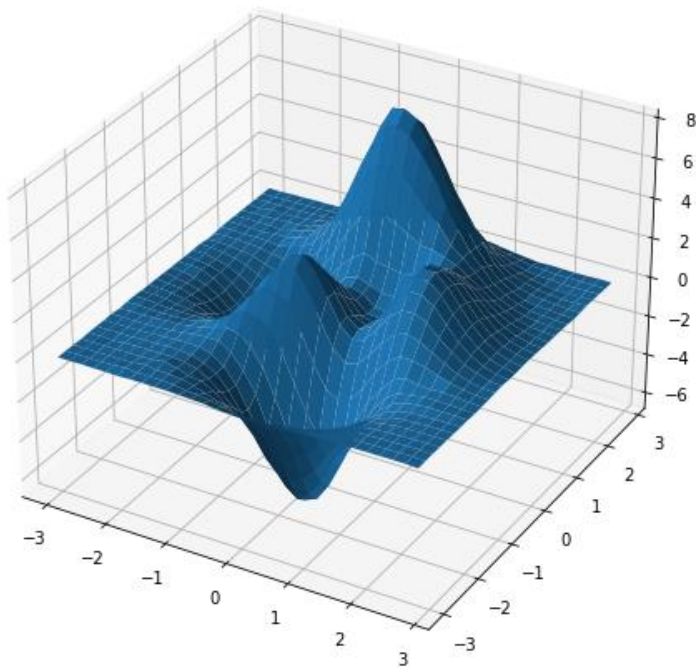
Se puede observar cómo la gráfica mejora considerablemente mediante aumenta el número de capas y el número de neuronas.

Partiendo de estos resultados, se puede concluir que las arquitecturas con más capas y más neuronas tienden a tener un menor error de generalización.

- **Realizando pruebas con diferentes épocas de entrenamiento:**

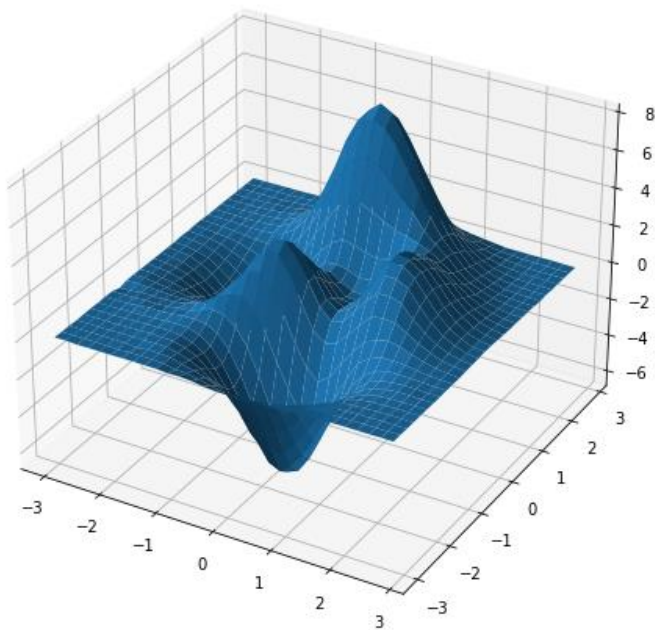
```
# Se crea la red neuronal de regresión  
net = MLPRegressor(solver = 'lbfgs', max_iter=10000, alpha = 1e-3, hidden_layer_sizes = (100,100,100))
```

Error de generalización: 0.005642310768935724



```
# Se crea la red neuronal de regresión  
net = MLPRegressor(solver = 'lbfgs', max_iter=20000, alpha = 1e-3, hidden_layer_sizes = (100,100,100))
```

Error de generalización: 0.004820432839315158



Se observa que incrementar el número de épocas de entrenamiento de 10000 a 20000 ha llevado a una mejora en el rendimiento de la red neuronal. El error de generalización se redujo de 0.005642310768935724 a 0.004820432839315158.

El aumento en el número de épocas permitió a la red neuronal ajustarse mejor a los datos de entrenamiento y lograr una mayor capacidad de generalización en la aproximación de la función.

Conclusión:

Por medio de estas pruebas se llegó a la conclusión de que la mejor solución para este caso en específico se da utilizando una arquitectura con más capas y más neuronas, además de aumentar el número de épocas de entrenamiento, para disminuir el error de generalización.

Para la solución se utilizó una arquitectura de tres capas, cada una de 100 neuronas y 20000 épocas de entrenamiento.