

```

import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

X = np.linspace(1, 100, 1000).reshape(-1, 1)
y = X**3 + 2 * X

model = Sequential([
    Dense(64, activation='relu', input_shape=(1,)),
    Dense(64, activation='relu'),
    Dense(1)
])

model.compile(optimizer='adam', loss='mse')
model.fit(X, y, epochs=200, batch_size=32, verbose=0)

y_pred = model.predict(X)

plt.scatter(y, y_pred, alpha=0.5)
plt.xlabel("Wartość oczekiwana")
plt.ylabel("Wartość prognozowana")
plt.title("f(x) = x³ + 2x")
plt.grid(True)
plt.show()

```

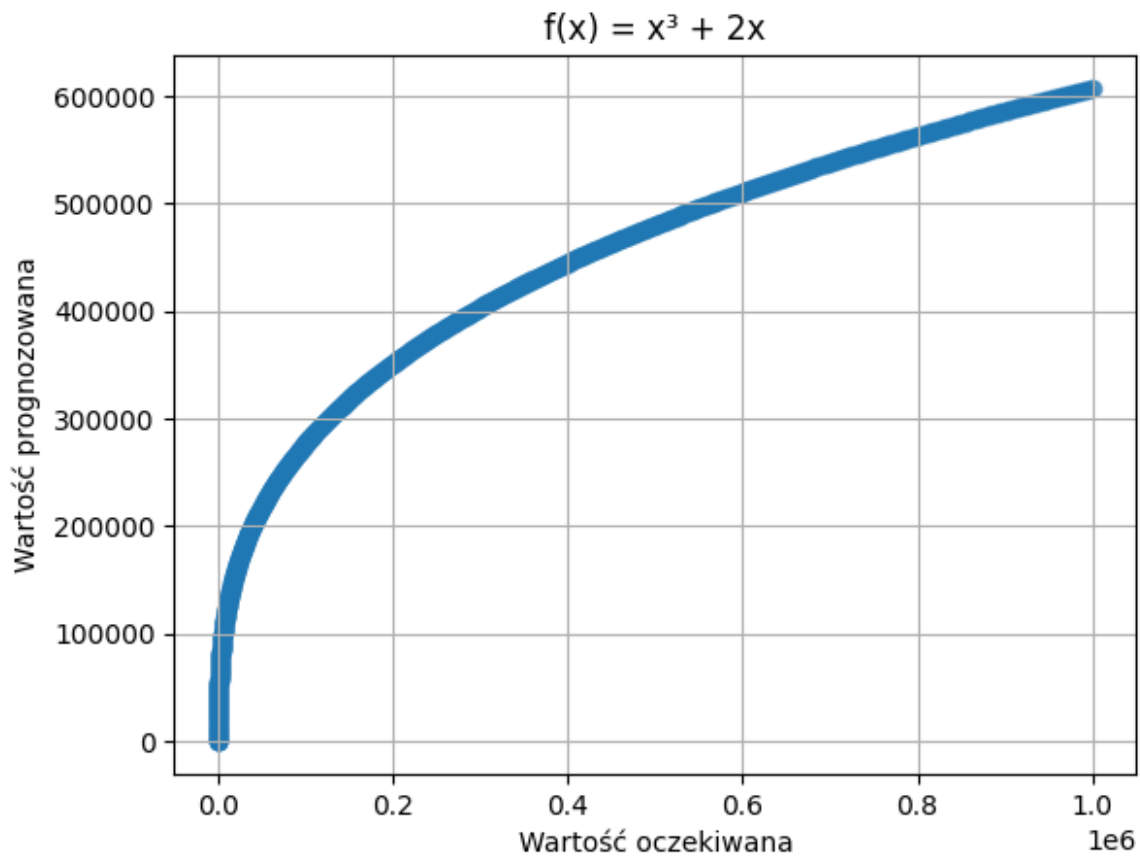
c:\Users\szymo\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\layers\core\dense.py:93: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

```

    super().__init__(activity_regularizer=activity_regularizer,
**kwargs)

```

32/32 ————— 0s 726us/step



```
X = np.random.rand(1000, 5) * [80, 40, 2, 3000, 10]
y = np.random.rand(1000, 3) * [200, 180, 120]

model = Sequential([
    Dense(64, activation='relu', input_shape=(5,)),
    Dense(64, activation='relu'),
    Dense(3)
])

model.compile(optimizer='adam', loss='mse')
model.fit(X, y, epochs=100, batch_size=32, verbose=0)

loss = model.evaluate(X, y, verbose=0)
print(f"Strata końcowa (MSE): {loss:.4f}")

Strata końcowa (MSE): 2688.4290
```