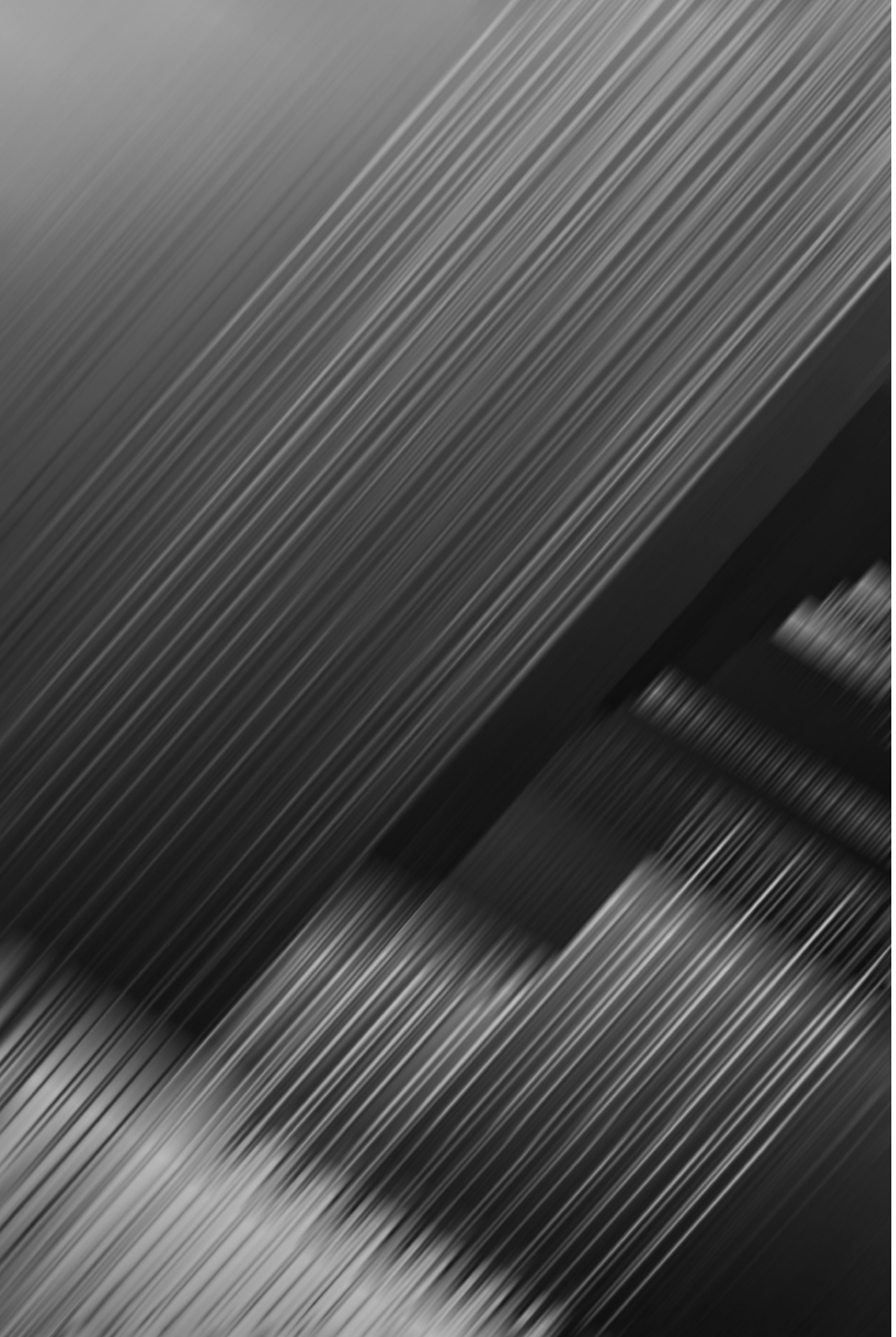


Deep Learning

Redes Neuronales con TensorFlow

Septiembre 1 – Septiembre 6

Regresión

Deep Learning

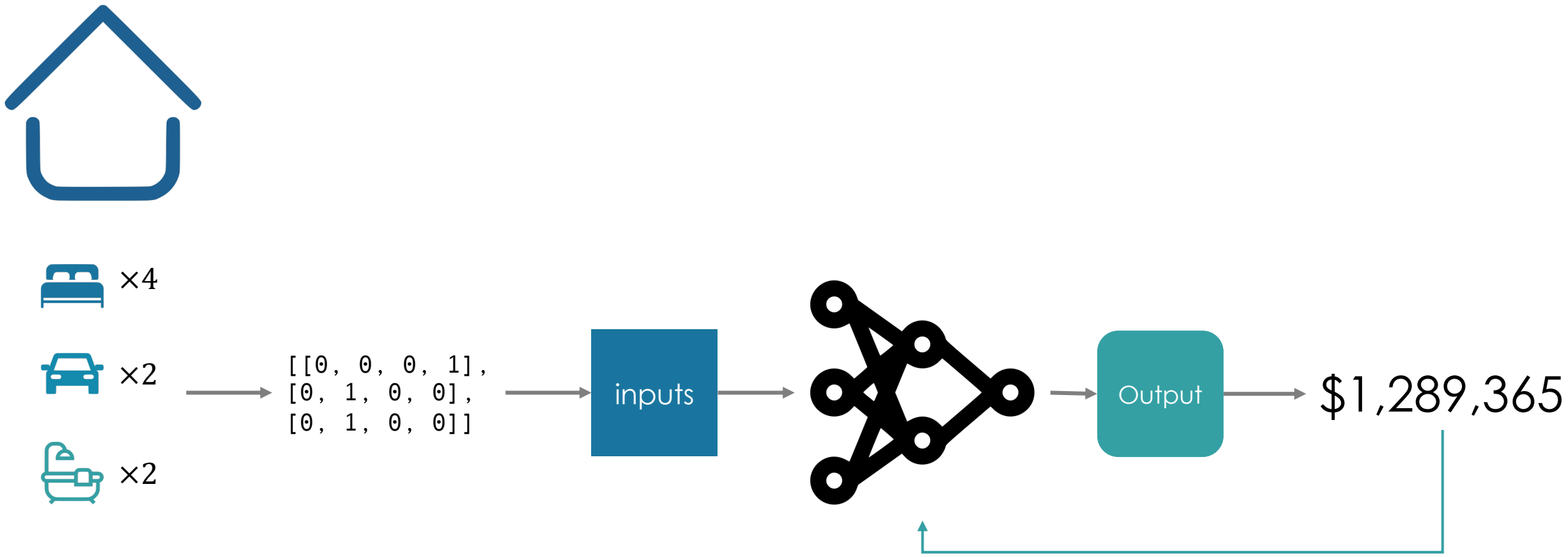
Redes Neuronales con TensorFlow

Septiembre 1 – Septiembre 6

Regresión

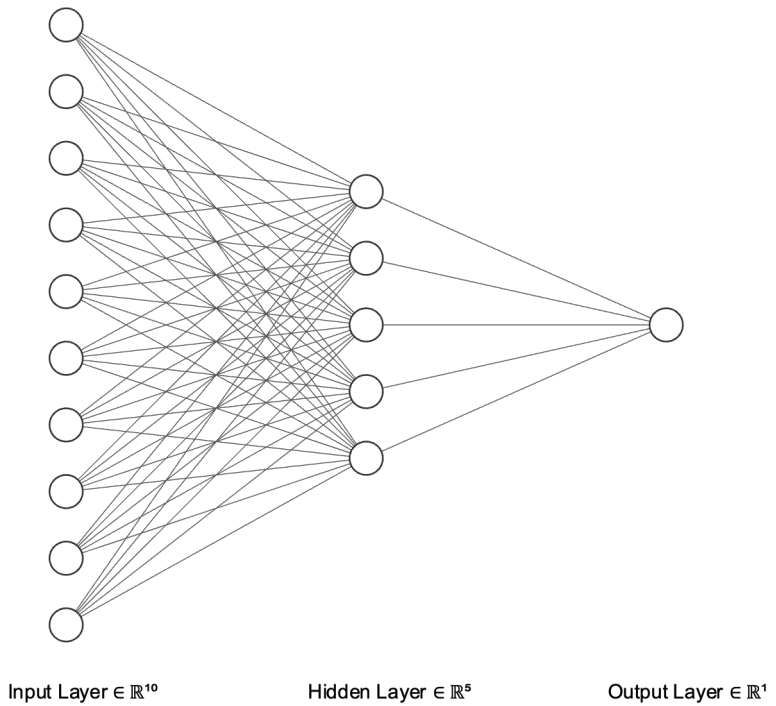
- Regresión Lineal
- Visualización de resultados
- Optimizadores
- Evaluación de modelos
- Regresión en TensorFlow

Regresión lineal



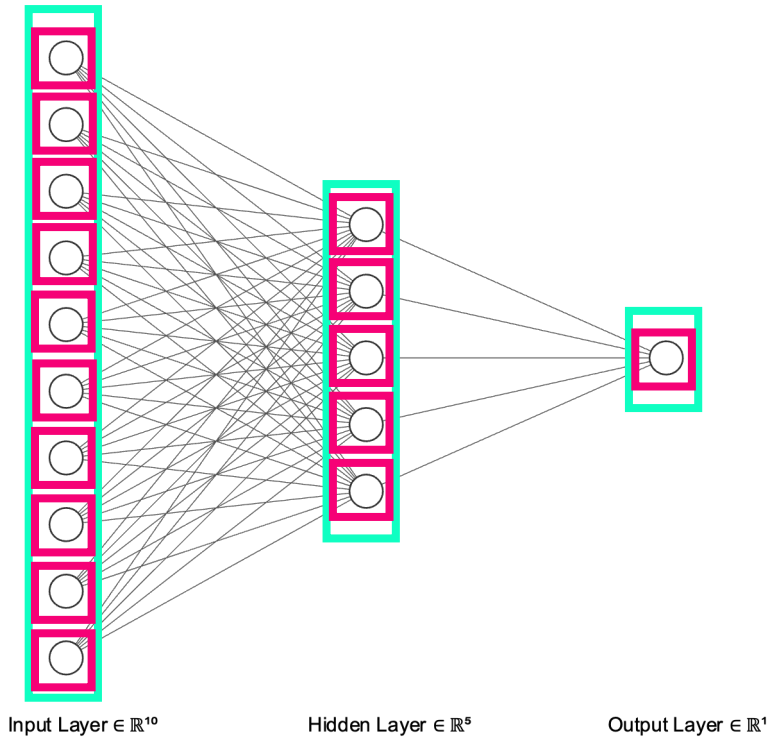
Código vs Concepto: Capas y neuronas

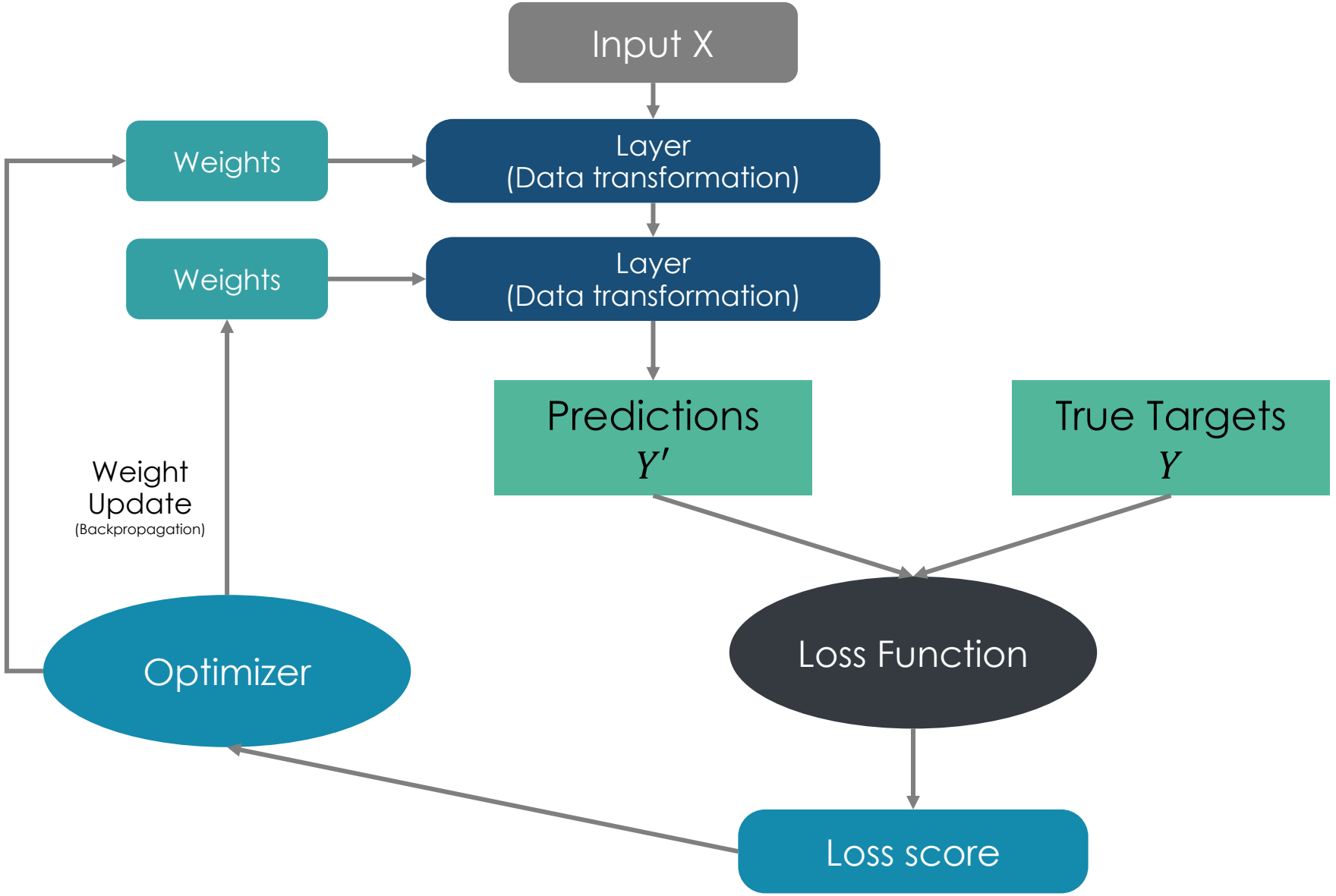
```
# random seed
tf.random.set_seed(42)

# Crear modelo
model = tf.keras.Sequential([
        tf.keras.layers.Dense(10)
        tf.keras.layers.Dense(5)
        tf.keras.layers.Dense(1)
])

# Compilar modelo
model.compile(loss=tf.keras.losses.mae,
        optimizer=tf.keras.optimizers.SGD(),
        metrics=["mae"])

# Fit model
model.fit(tf.expand_dims(X, axis=-1), y, epochs=5)
```



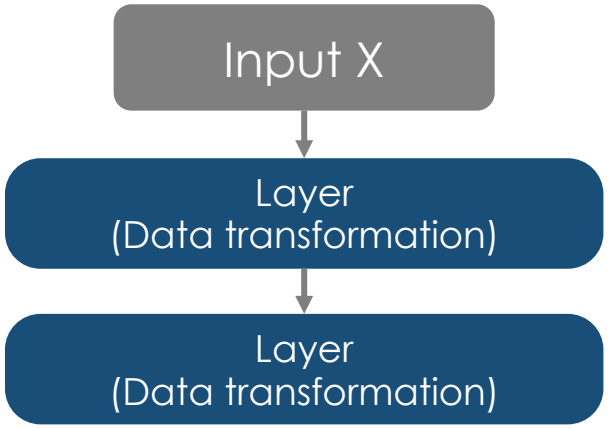
Código vs Concepto: Capas y neuronas

```python
# random seed
tf.random.set_seed(42)

# Crear modelo
model = tf.keras.Sequential([
        tf.keras.layers.Dense(10)
        tf.keras.layers.Dense(5)
        tf.keras.layers.Dense(1)
])

# Compilar modelo
model.compile(loss=tf.keras.losses.mae,
        optimizer=tf.keras.optimizers.SGD(),
        metrics=["mae"])

# Fit model
model.fit(tf.expand_dims(X, axis=-1), y, epochs=5)
```



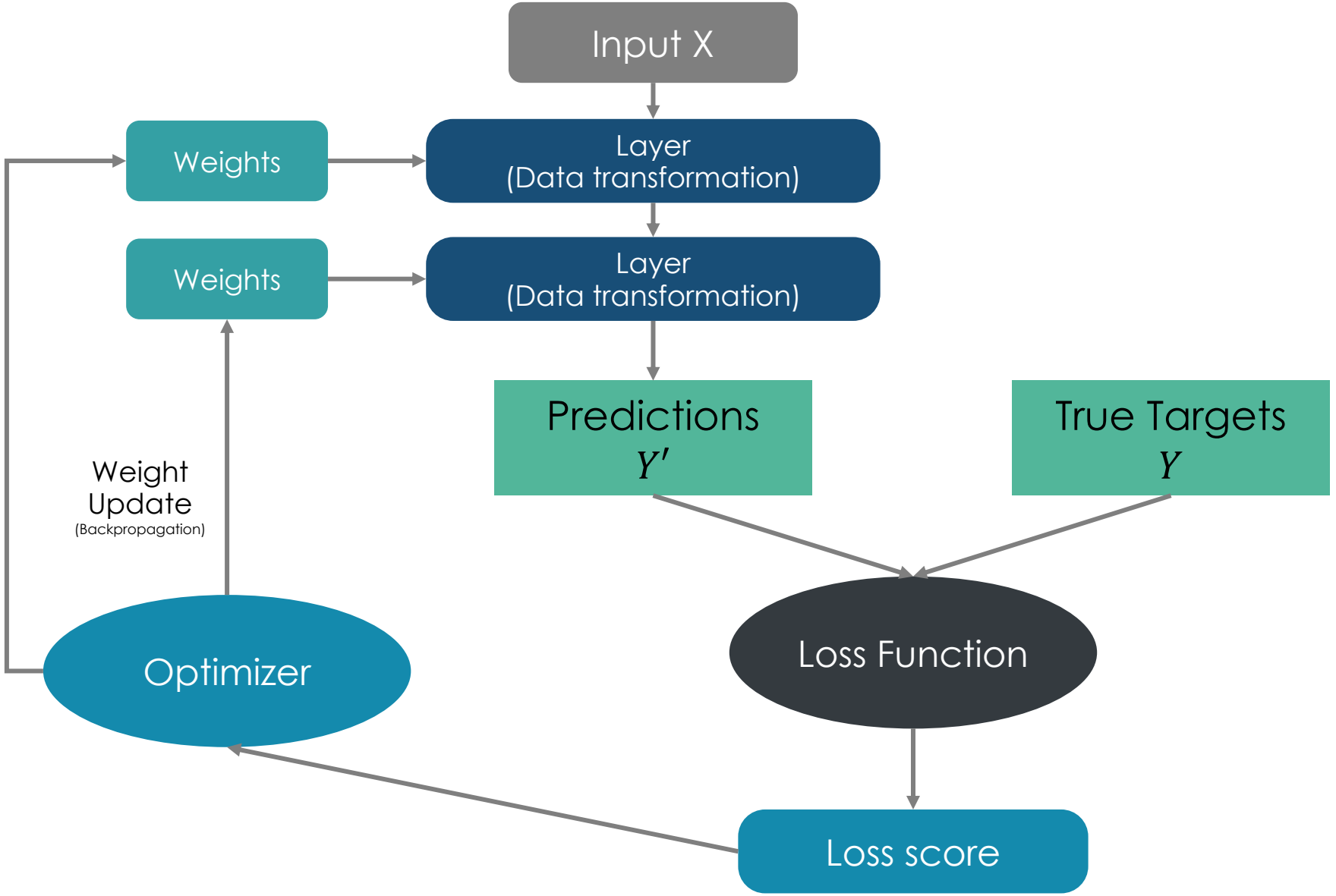


*Tomado de Francois Chollet (2021)

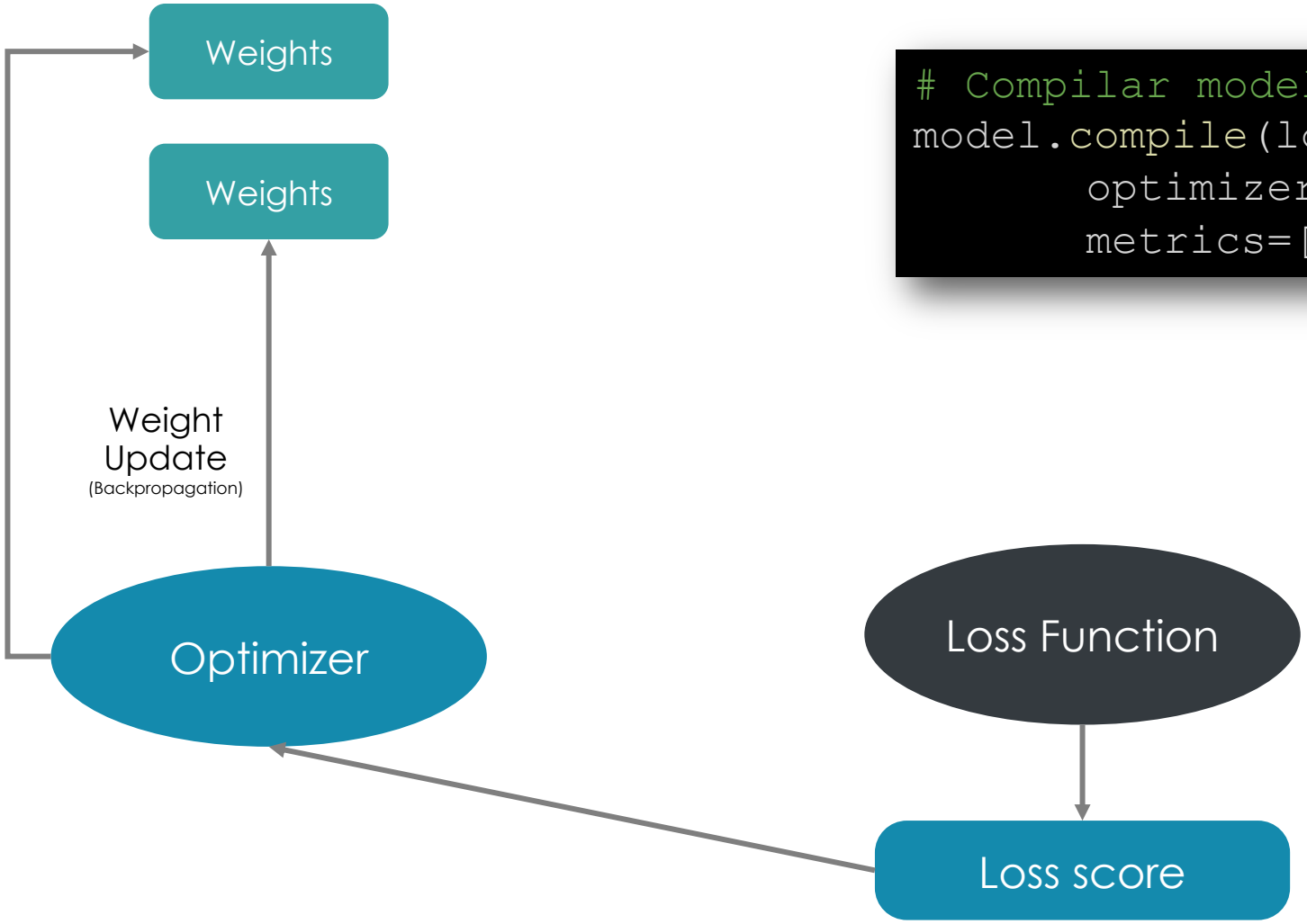



```python
# random seed
tf.random.set_seed(42)

# Crear modelo
model = tf.keras.Sequential([
        tf.keras.layers.Dense(10)
        tf.keras.layers.Dense(5)
        tf.keras.layers.Dense(1)
])
```

*Tomado de Francois Chollet (2021)

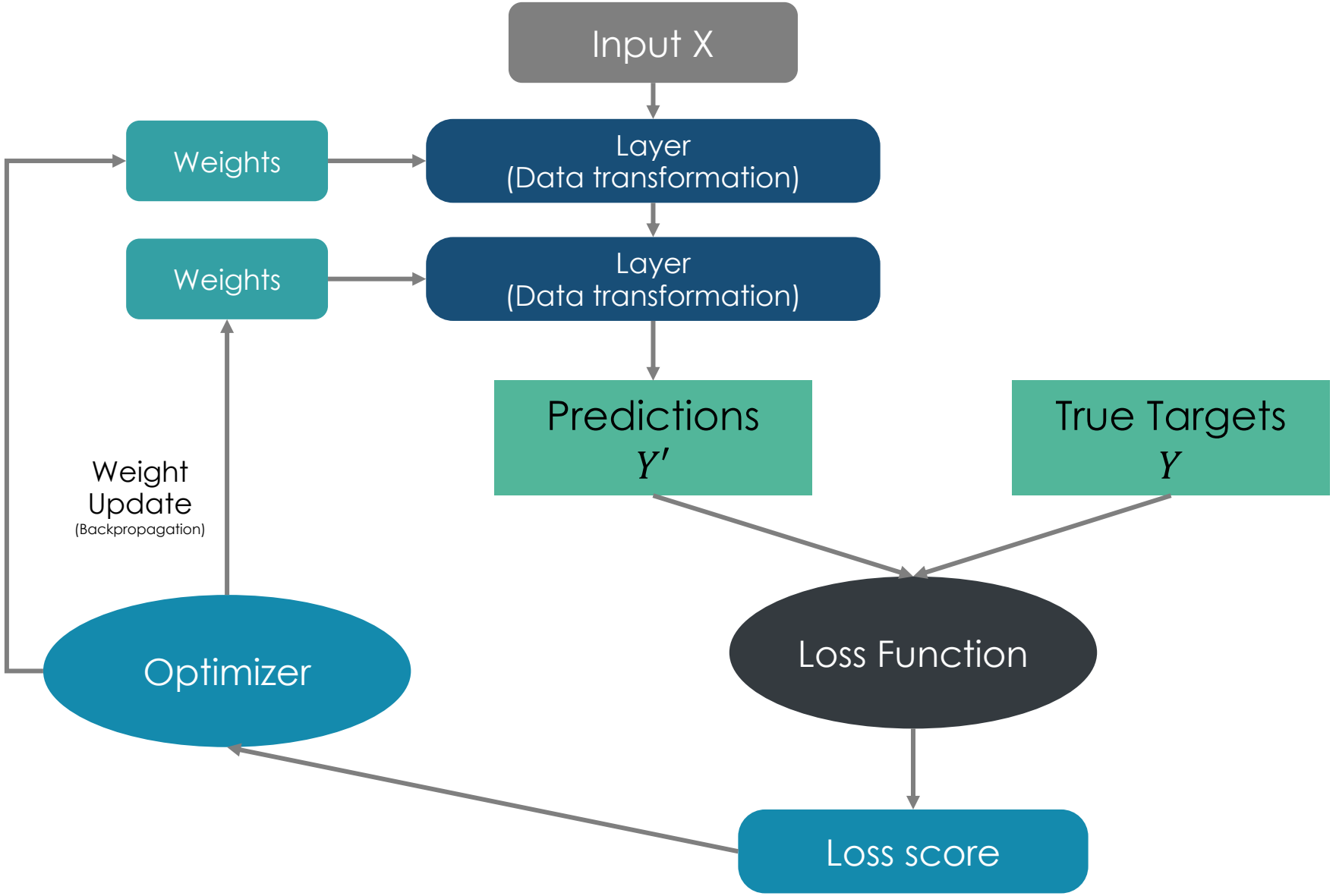


*Tomado de Francois Chollet (2021)

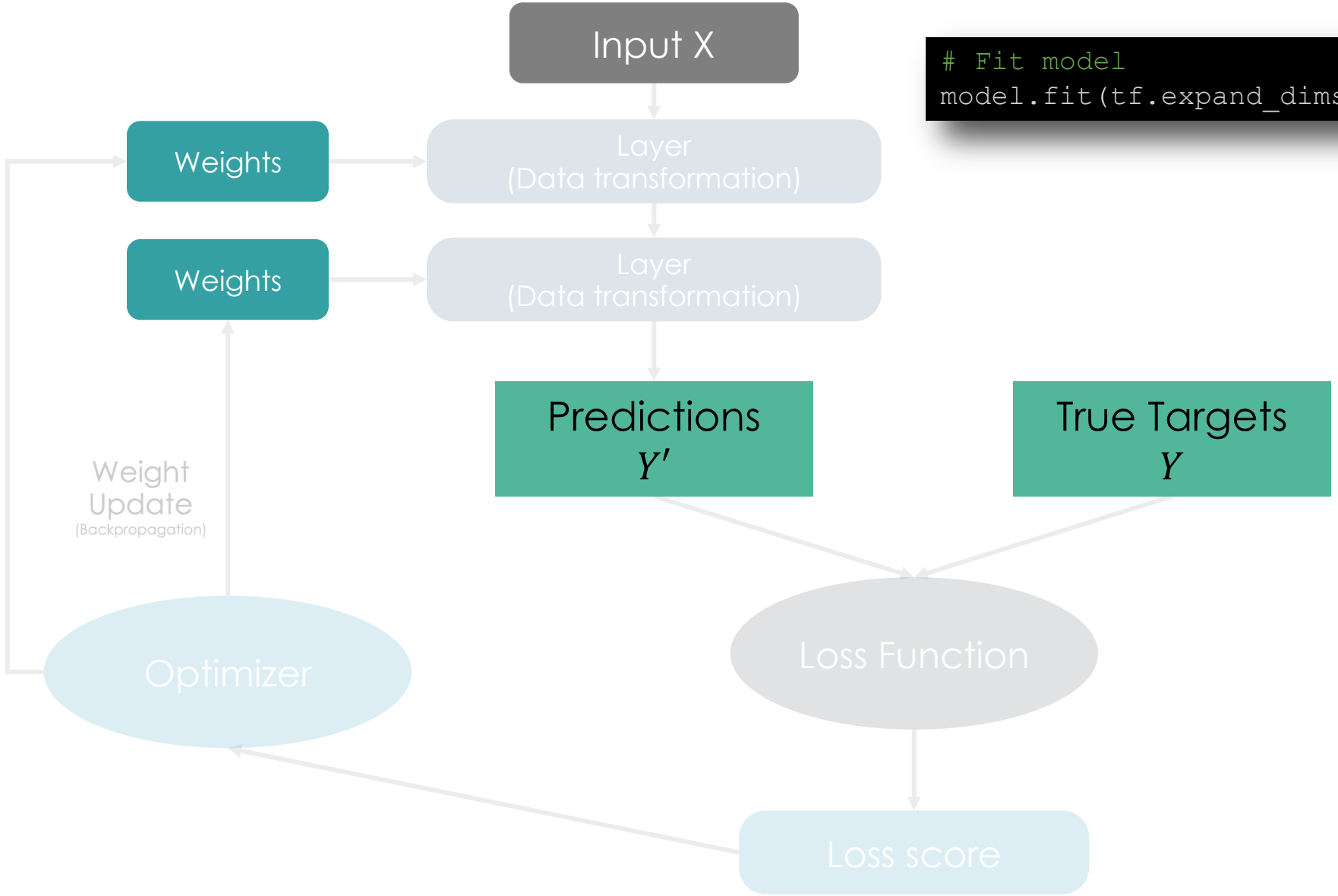



```
# Compilar modelo
model.compile(loss=tf.keras.losses.mae,
              optimizer=tf.keras.optimizers.SGD(),
              metrics=["mae"])
```

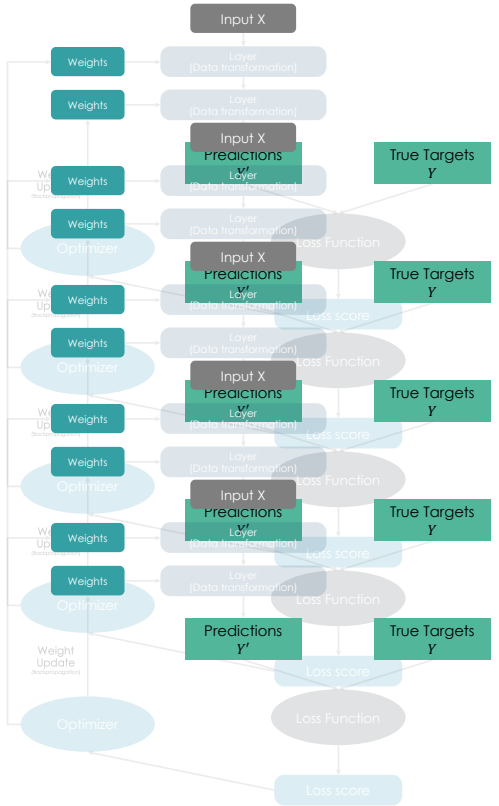
*Tomado de Francois Chollet (2021)



*Tomado de Francois Chollet (2021)



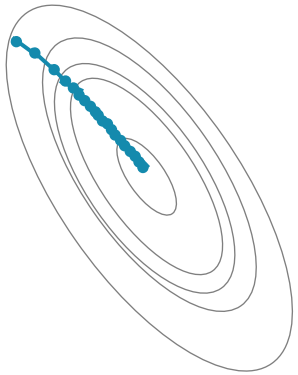

```
# Fit model
model.fit(tf.expand_dims(X, axis=-1), y, epochs=5)
```



*Tomado de Francois Chollet (2021)

Optimizadores

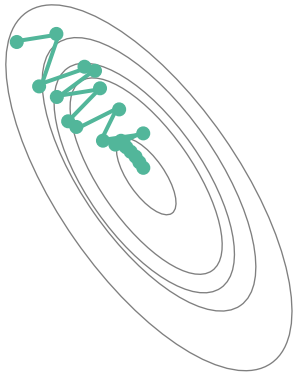
Gradient Descent



Suavizado pero muy lento

- Se puede usar con una paralelización pero depende de la longitud del dataset
- Es eficiente

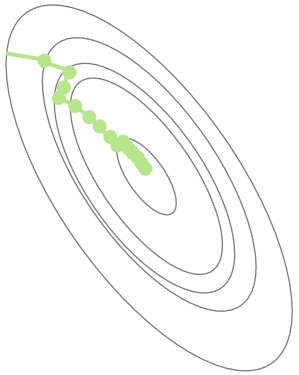
Stochastic Gradient Descent



Ruidoso pero lento

- Usualmente converge más rápido que el Gradient Descent en datasets amplios
- No se necesita todo el dataset para tener una aproximación precisa
- No ocupa mucha memoria
- Se estima después de cada batch de datos de entrenamiento, lo que ocasiona que haya saltos

Adam



Suavizado y rápido

- Regla No.1: Usa Adam
- Los hiperparámetros dados del optimizador no necesitan ser ajustados
- Realiza el aprendizaje a través de "pasos adaptativos"
- De los tres, es el que más usa memoria

Referencias

- Foto de portada: D_21 Gallery
- Meor Amer, (2022), **“A visual Introduction to Deep Learning”**
- Roy Keyes (2022), **“Deep Learning”**

GRACIAS

emilio.sandpal@gmail.com

milioe