

OPTIMIZER NEURAL NETWORKS

Presented by:
Madhusudhanan Jeyaram

Overview

- 1.** What is Optimizers
- 2.** Why Do We Need It
- 3.** Where Does It Work?
- 4.** When to Use What?
- 5.** Sample Code(Keras)
- 6.** Reference

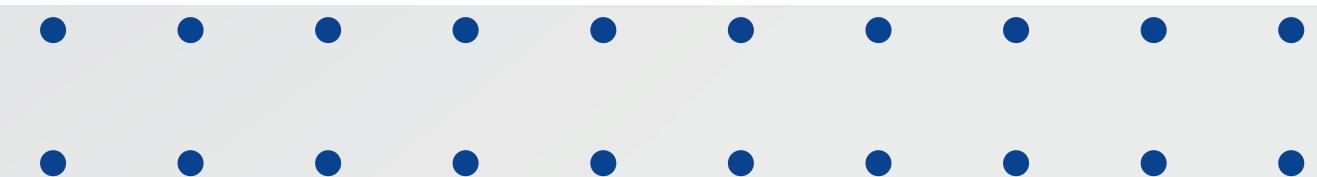


What is Optimizers

An optimizer is like a friendly coach who helps your deep learning program learn from its mistakes.

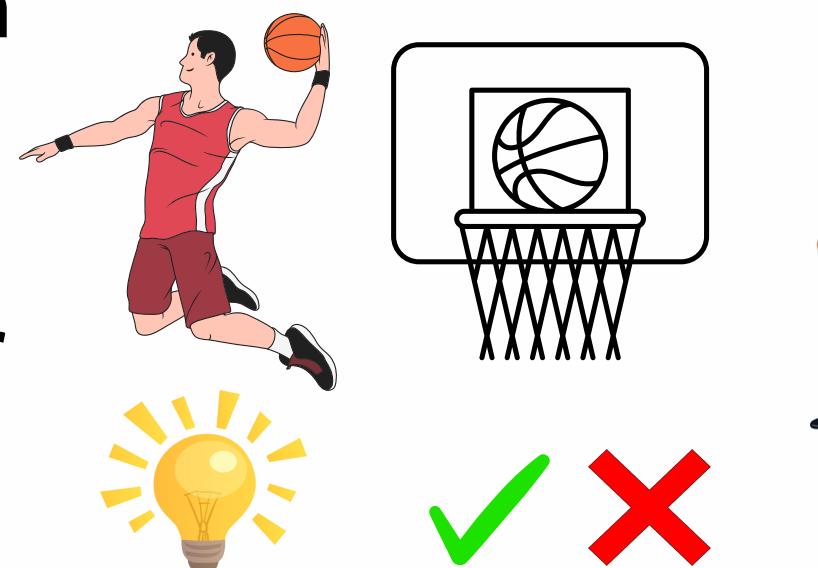


Guess → **Learn** → **Improve**



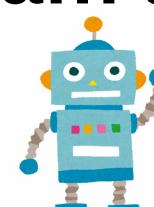
Why Do We Need It

A deep learning program doesn't start smart – it makes random guesses, like shooting a basketball with eyes closed.

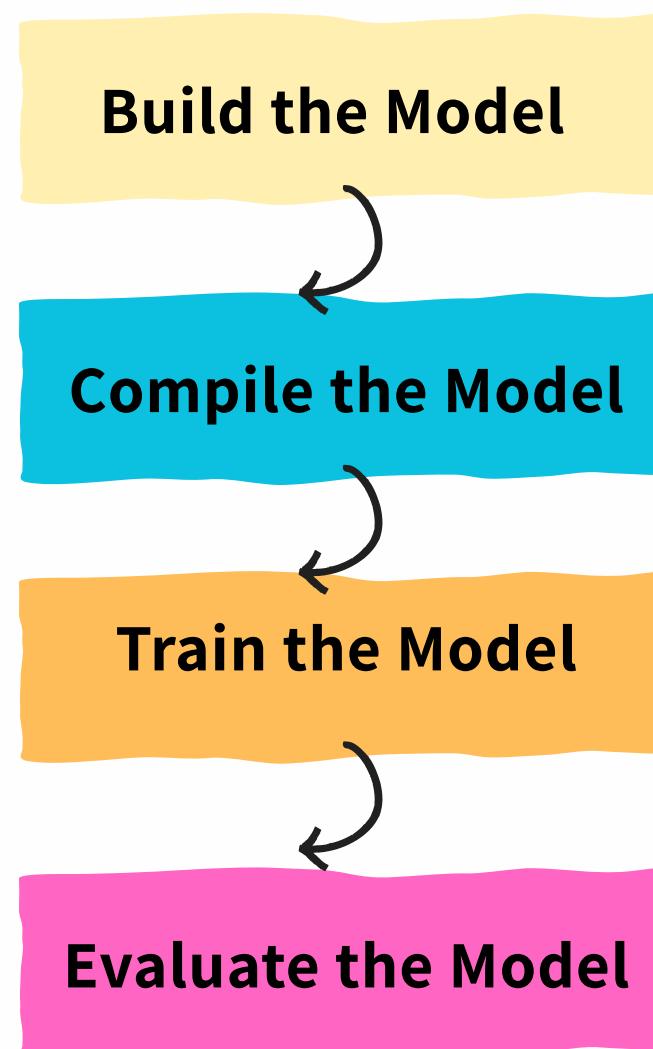


The optimizer acts like a coach who helps it aim better each time.

Bit by bit, the program learns from mistakes and becomes smarter.



Where Does It Work?



**Optimizers work with all
types of neural networks
in deep learning**

Presented by: Madhusudhanan Jeyaram

<https://madhusudhanan.online/>

When to Use What?

Deep Learning Optimizer Reference

Optimizer	Learning Rate	Best For / Model Type	When to use
 SGD (Stoch... ▾	0.01 Defa... ▾	Simple models, linear regression	basics / small datasets
 Adam (Adapti... ▾	0.01 Defa... ▾	CNNs, RNNs, Transformers	Default choice for deep learning
 RMSprop (Ro... ▾	0.01 Defa... ▾	RNN / LSTM / time-series models	For RNN/LSTM and time-series models
 Optimizer De... ▾	0.01 Defa... ▾	Sparse or NLP data	For sparse or text data (NLP)
 AdamW (Ada... ▾	0.01 Defa... ▾	Large transformer-based or overfitting models	For large models / to reduce overfitting

Presented by: Madhusudhanan Jeyaram

<https://madhusudhanan.online/>

Sample Code (Keras)

```
● ● ●

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

# Step 1: Create sample data (X = input, y = target)
import numpy as np
X = np.array([[1], [2], [3], [4]], dtype=float)
y = np.array([[2], [4], [6], [8]], dtype=float) # y = 2x

# Step 2: Build a simple model
model = keras.Sequential([
    layers.Dense(units=1, input_shape=[1])
])

# ---- CASE 1: Without optimizer (you'll get stuck!) ----
# model.compile(loss='mean_squared_error') # ❌ no optimizer
# model.fit(X, y, epochs=10) # this won't improve

# ---- CASE 2: With optimizer (the brain of learning!) ----
model.compile(optimizer='adam', loss='mean_squared_error')

# Step 3: Train the model
model.fit(X, y, epochs=500, verbose=0)

# Step 4: Test it
print("Prediction for 5:", model.predict([[5]]))
```

THANK YOU

Presented by:
Madhusudhanan Jeyaram