



**disruptive**  
digital  
education

# deep learning com tensorflow

programa | bootcamp remote learning

# deep learning com tensorflow

Este programa foi concebido por profissionais da área com uma vasta experiência nacional e internacional.

## O QUE É?

O Deep Learning é uma área da Inteligência Artificial focada na criação de modelos capazes de aprender a partir de grandes volumes de dados, utilizando redes neurais inspiradas no funcionamento do cérebro humano.

Esta área permite desenvolver soluções para problemas complexos como reconhecimento de imagem, processamento de linguagem natural, sistemas de recomendação e automação inteligente, sendo hoje uma das bases tecnológicas por trás de ferramentas como assistentes virtuais, motores de pesquisa e modelos generativos.

**duração total** 32h

### investimento

375,00€ x 2 (sem juros)  
(pronto pagamento: 5% de desconto)

### ADMISSÃO

Idade Mínima de 18 anos.

Conhecimentos básicos de programação em Python (variáveis, loops, funções). Noções de álgebra linear e estatística descritiva são recomendáveis mas não obrigatórias — os conceitos matemáticos necessários são revistos durante o curso.

## **SOBRE O BOOTCAMP**

Este curso oferece uma formação prática e progressiva em Deep Learning utilizando TensorFlow 2.x e Keras. Partindo dos fundamentos de Machine Learning — regressão linear e classificação — os formandos avançam até arquiteturas avançadas como CNNs, RNNs, Transformers e GANs, culminando no deployment de modelos em produção. Cada sessão combina exposição teórica com exercícios práticos em datasets reais.

## **OBJETIVOS**

No final do curso, vais ser capaz de implementar algoritmos de Machine Learning como regressão linear e logística, compreender a base matemática e a lógica das redes neurais e do backpropagation, e construir, treinar e avaliar modelos de Deep Learning com TensorFlow e Keras. Vais também aprender a melhorar a performance dos modelos com técnicas de regularização e otimização, trabalhar com CNNs e Transfer Learning em visão computacional, e processar dados como texto e séries temporais com RNNs, LSTMs e GRUs. Além disso, vais compreender conceitos mais avançados como mecanismos de atenção, Transformers e o impacto dos LLMs, implementar modelos generativos como Autoencoders e GANs, e preparar modelos para produção com ferramentas como TF Serving, TF Lite e Docker.

## **PERFIL**

Conhecimentos básicos de programação em Python (variáveis, loops, funções). Noções de álgebra linear e estatística descritiva são recomendáveis mas não obrigatórias — os conceitos matemáticos necessários são revistos durante o curso.

## CONTEÚDO PROGRAMÁTICO

1.

### Fundamentos de Machine Learning: Regressão e Classificação

- › Implementar algoritmos de Machine Learning clássico: regressão linear e regressão logística
- › Compreender a matemática e a intuição por trás das redes neurais e do backpropagation
- › Construir, treinar e avaliar modelos de Deep Learning com TensorFlow 2.x e Keras
- › Aplicar técnicas de regularização e otimização para melhorar a performance dos modelos
- › Utilizar CNNs e Transfer Learning para problemas de visão computacional
- › Processar texto e séries temporais com RNNs, LSTMs e GRUs
- › Compreender o mecanismo de atenção, Transformers e o impacto dos LLMs
- › Implementar modelos generativos: Autoencoders e GANs
- › Exportar e fazer deploy de modelos com TF Serving, TF Lite e Docker

2.

### Estrutura, Funcionalidades e Potencial da Ferramenta

- › O perceptrão: modelo biológico vs modelo computacional
- › Funções de ativação: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax — quando usar cada uma
- › Arquitetura de uma rede neural: camadas de entrada, ocultas e de saída
- › Forward propagation: do input à previsão, passo a passo
- › Funções de custo: MSE para regressão, Binary Cross-Entropy e Categorical Cross-Entropy
- › Backpropagation: a regra da cadeia aplicada ao treino de redes neurais
- › Gradient descent e as suas variantes: batch, mini-batch e stochastic (SGD)
- › Introdução ao TensorFlow 2.x: tensores, eager execution, tf.GradientTape
- › Primeira rede neural com Keras: Sequential API, Dense layers, compile e fit

## CONTEÚDO PROGRAMÁTICO

### 3. Treino de Redes Profundas: Regularização e Otimização

- › Problemas de redes profundas: vanishing e exploding gradients
- › Inicialização de pesos: Xavier/Glorot e He — como evitar gradientes instáveis
- › Otimizadores avançados: SGD com momentum, RMSprop, Adam, AdaGrad — comparação visual e prática
- ▶ Learning rate scheduling: step decay, exponential decay, cosine annealing e ReduceLRonPlateau
- ▶ Regularização L1 e L2 (weight decay): intuição geométrica e impacto nos pesos
- ▶ Dropout: conceito, taxa ideal e como aplicar em Keras
- ▶ Batch Normalization: normalizar ativações para estabilizar e acelerar o treino
- ▶ Early Stopping e callbacks no Keras: ModelCheckpoint, EarlyStopping, TensorBoard
- ▶ Data augmentation: transformações geométricas e de cor para aumentar o dataset
- ▶ TensorBoard: monitorização de loss, accuracy, histogramas de pesos e grafos do modelo

### 4. Redes Neurais Convolucionais (CNNs) para Visão Computacional

- › Intuição das convoluções: filtros como detetores de features (bordas, texturas, formas)
- › Camadas fundamentais: Conv2D, MaxPooling, AveragePooling, GlobalAveragePooling e Flatten
- › Hiperparâmetros: kernel size, stride, padding (same vs valid) e número de filtros
- › Arquiteturas clássicas: LeNet-5, AlexNet, VGG-16 — evolução histórica e decisões de design
- › ResNet e skip connections: como resolver o degradation problem em redes muito profundas
- › Transfer Learning: reutilizar modelos pré-treinados (ImageNet) para novos problemas
- › Fine-tuning: estratégias de descongelamento progressivo de camadas
- › tf.data API: pipelines eficientes com prefetch, cache, shuffle e batch

## CONTEÚDO PROGRAMÁTICO

5.

### Redes Neurais Recorrentes (RNNs) & Processamento de Sequências

- › Dados sequenciais: texto, séries temporais e sinais — porque é que a ordem importa
- › RNN básica: unfolding, hidden state e o problema da memória de curto prazo
- › LSTM (Long Short-Term Memory): cell state, gates (forget, input, output) e fluxo de informação
- › GRU (Gated Recurrent Unit): simplificação do LSTM — arquitetura e quando usar cada um
- › Bidirectional RNNs e stacked layers: capturar contexto nos dois sentidos
- › Pré-processamento de texto: tokenização, padding, vocabulário e sequências de inteiros
- › Word Embeddings: Word2Vec, GloVe e `tf.keras.layers.Embedding`
- › Previsão de séries temporais: janelas deslizantes, normalização e walk-forward validation

6

### Arquiteturas Avançadas: Transformers, Autoencoders e GANs

- › Limitações das RNNs: processamento sequencial e dificuldade com dependências longas
- › Mecanismo de Atenção: intuição (query, key, value), score functions e self-attention
- › Arquitetura Transformer: encoder-decoder, multi-head attention e positional encoding
- › Visão geral de BERT, GPT e LLMs: como os Transformers mudaram o NLP e a IA
- › Autoencoders: arquitetura encoder-decoder para compressão e reconstrução de dados
- › Variational Autoencoders (VAEs): espaço latente contínuo e geração de novos dados
- › GANs (Generative Adversarial Networks): o jogo entre generator e discriminator
- › DCGAN: geração de imagens com convoluções — arquitetura e técnicas de estabilização

## CONTEÚDO PROGRAMÁTICO

### **7.** Deploy de Modelos & Introdução a MLOps

- › Do notebook à produção: desafios reais de deployment (latência, escala, monitorização)
- › Salvamento de modelos: SavedModel format, HDF5, TF Checkpoints e versionamento
- › TensorFlow Serving: servir modelos via REST API e gRPC
- › TensorFlow Lite: otimização para dispositivos móveis e edge computing
- › TensorFlow.js: execução de modelos no browser — casos de uso e limitações
- › Otimização de modelos: quantização (post-training e quantization-aware) e pruning
- › Conceitos de MLOps: versionamento de dados e modelos, CI/CD para ML, monitorização de drift
- › Introdução ao TFX (TensorFlow Extended): componentes de um pipeline ML em produção
- › Docker básico: containerizar um modelo para deployment reproduzível

### **8** Projeto Final: Aplicação End-to-End de Deep Learning

- › Definição do problema e seleção do dataset (orientação individualizada pelo formador)
- › Análise exploratória dos dados (EDA) e pré-processamento
- › Seleção e justificação da arquitetura: CNN, RNN, Transformer ou combinação
- › Treino com regularização, otimização e monitorização via TensorBoard
- › Avaliação com métricas adequadas ao problema e análise de erros (confusion matrix, exemplos mal classificados)
- › Exportação do modelo para deployment (SavedModel ou TFLite)
- › Apresentação final: pitch técnico de 10 minutos por aluno/grupo
- › Feedback bidirecional, discussão de melhorias e próximos passos

## EQUIPAMENTO

Um computador com ligação a internet;  
Webcam e microfone;  
Documentação em formato digital.

## CERTIFICAÇÃO

A certificação é entregue aos alunos em formato digital e inclui: carga horária, modular e total do curso, temáticas, notas finais de cada módulo e média final do curso, certificação DGERT, carimbo e assinatura (da coordenação pedagógica do programa ou administração da escola).

## REMOTE LEARNING

Esta formação irá manter e garantir a qualidade dos cursos presenciais, uma vez que as aulas em formato remote learning decorrem num ambiente de sala de aula virtual, com formação live, permitindo assim que todos os participantes se conheçam e interajam entre si, adquiram conhecimento independentemente da sua localização e beneficiem de feedback em tempo real.

A formação será estruturada por sessões em direto transmitidas a partir de estúdios em Lisboa e no Porto, devidamente equipados tecnicamente com tecnologia broadcast ao nível de áudio, luz e imagem multicâmara. O acesso às aulas, conteúdos e documentação de apoio será feito através de uma plataforma, aliada a um sistema de vídeo e áudio conferência online com interação bidirecional entre os tutores e formandos.

## Notas

- › Em caso de cancelamento do Bootcamp por parte da EDIT., o valor pago pelo(a) participante antecipadamente será reembolsado na sua íntegra. Caso o cancelamento seja efetuado pelo(a) aluno(a) será devido uma multa (cláusula penal) de 10% (dez por cento) do valor pago. Não será restituído o valor da inscrição após a realização do workshop.
- › A EDIT., livre de quaisquer ônus para com o(a) aluno(a), poderá utilizar a sua imagem e som de voz para fins de divulgação do workshop e das suas atividades, podendo, para tanto, reproduzi-lo ou divulgá-lo nos seus canais de comunicação.
- › A EDIT. reserva-se o direito de alterar datas, horários, locais ou mesmo cancelar workshops de acordo com um número mínimo de interessados. Nesse caso, todos os inscritos serão previamente avisados, até 24 horas antes do início do workshop.
- › Ficarão ao critério da EDIT. os cancelamentos dos Bootcamps por outros motivos que aqui não foram informados.
- › Demais valores gastos pelo(a) aluno(a) não serão reembolsáveis, como valores gastos com alojamento, hotel, passagem de qualquer tipo e deslocações.



**disruptive**  
digital  
education

[weareedit.io](http://weareedit.io)

**EDIT.**  
**Lisboa**

Alameda D. Afonso Henriques, 7A  
1900-178 Lisboa

(+ 351) 210 182 455  
(chamada para rede fixa nacional)  
[geral@edit.com.pt](mailto:geral@edit.com.pt)

**EDIT.**  
**Porto**

Rua Alferes Malheiro, 226  
4000-057 Porto

(+ 351) 224 960 345  
(chamada para rede fixa nacional)  
[geral@edit.com.pt](mailto:geral@edit.com.pt)