

# TP558 - Tópicos avançados em Machine Learning: *Introdução ao curso*



# Sobre o curso

- Curso prático que oferecerá uma visão ampla e detalhada sobre diferentes algoritmos avançados de aprendizado de máquina (ML) e suas aplicações em diferentes áreas do conhecimento.
- Cobriremos algoritmos como
  - Redes generativas,
  - Redes recorrentes,
  - Redes de atenção,
  - Redes grafo neurais,
  - Aprendizado por reforço profundo,
  - Autoencoders,
  - Transformers,
  - Detectores de objetos,
  - etc.

# Sobre o curso

- **Dinâmica do curso**

- O curso será *dividido em vários seminários preparados e apresentados pelos alunos*, cada um cobrindo um tipo diferente de algoritmo.
- Ao final de cada seminário, os *alunos deverão responder a um quiz*, preparado pelo apresentador, sobre o algoritmo apresentado.
- Ao final do curso, os *alunos deverão apresentar um projeto final*, incluindo um *relatório em formato de artigo científico*, envolvendo a aplicação de um algoritmo avançado de ML a um problema de sua escolha (*de preferência, alinhado com sua pesquisa*).

# Sobre o curso

- **Objetivo principal do curso**

- Ao final do curso, os alunos devem ser capazes de entender e aplicar na prática os diferentes algoritmos estudados.

- **Pré-requisitos:**

- Disciplinas: TP555 ou TP557;
- Conceitos de ***álgebra linear*** (e.g., matrizes e vetores), ***cálculo*** (e.g., algoritmos de otimização, como o gradiente descendente), ***probabilidade e estatística*** (e.g., distribuições de probabilidade, média, desvio padrão, validação cruzada);
- Conhecimento intermediário/avançado de programação em Python.
- Bibliotecas: TensorFlow, PyTorch, Optuna/KerasTuner, SciKit-Learn e Pandas.

# Sobre o curso

- **Repositório**

- Criem um repositório público no github para armazenar o material gerado durante todo o curso, e.g., materiais dos seminários, proposta de projeto, projeto final, etc.
- Criem pastas distintas para cada seminário, para o projeto final e qualquer outro material que seja gerado durante o curso.
- Enviem o endereço do repositório para o professor via email.

# Sobre o curso

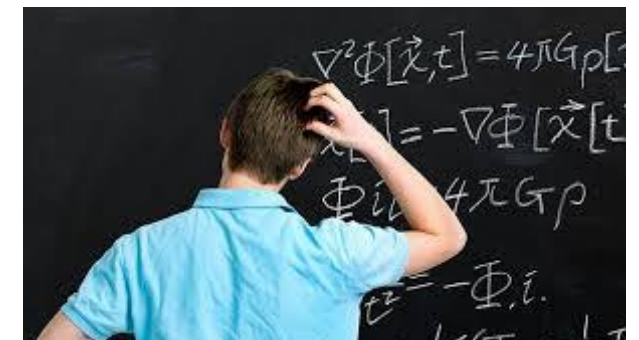


## Avaliações

- Seminários (S): 50%
  - Cada aluno irá estudar alguns algoritmos e apresentar aos demais.
  - Uma a duas semanas para estudo, preparação da apresentação e quiz.
- Quizzes (Q): 10%
  - Resolução dos quizzes preparados pelos alunos.
- Proposta do projeto final (PPF): 10%
  - Projeto prático com tema escolhido pelos alunos.
  - Um aluno por projeto.
  - Deve ser entregue na metade do segundo mês.
- Projeto final (PF): 30%
  - Apresentação na última semana de aula.
  - Entrega de relatório em formato de artigo científico.



$$NF = S*50\% + Q*10\% + PPF*10\% + PF*30\%$$



# Sobre o curso

- Instruções detalhadas sobre os seminários e projeto final.
  - Seminários: <http://tinyurl.com/tp558-seminars>
  - Projeto final: <http://tinyurl.com/tp558-final-project>

# Temas dos seminários

- Generative deep learning: Image Generation
  - Deep convolutional generative adversarial network (DCGAN)
    - ✓ <https://arxiv.org/abs/1511.06434>
    - ✓ [https://keras.io/examples/generative/dcgan\\_overriding\\_train\\_step/](https://keras.io/examples/generative/dcgan_overriding_train_step/)
  - Conditional GAN (CGAN)
    - ✓ <https://arxiv.org/abs/1411.1784>
    - ✓ [https://keras.io/examples/generative/conditional\\_gan/](https://keras.io/examples/generative/conditional_gan/)
  - Variational Autoencoder (VAE)
    - ✓ <https://arxiv.org/abs/1312.6114>
    - ✓ <https://www.tensorflow.org/tutorials/generative/cvae?hl=en>



# Temas dos seminários

- Generative deep learning: Image Generation
  - OOTDiffusion: Outfitting Fusion based Latent Diffusion for Controllable Virtual Try-on
    - ✓ <https://arxiv.org/abs/2403.01779>
    - ✓ <https://paperswithcode.com/paper/ootdiffusion-outfitting-fusion-based-latent>
  - V3D: Video Diffusion Models are Effective 3D Generators
    - ✓ <https://arxiv.org/abs/2403.06738>
    - ✓ <https://paperswithcode.com/paper/v3d-video-diffusion-models-are-effective-3d>

# Temas dos seminários

- Computer Vision: Image classification
  - Vision Transformer
    - ✓ <https://arxiv.org/abs/2010.11929>
    - ✓ [https://keras.io/examples/vision/image\\_classification\\_with\\_vision\\_transformer/](https://keras.io/examples/vision/image_classification_with_vision_transformer/)
  - Swin Transformer
    - ✓ <https://arxiv.org/abs/2103.14030>
    - ✓ [https://keras.io/examples/vision/swin\\_transformers/](https://keras.io/examples/vision/swin_transformers/)
  - Reptile
    - ✓ <https://arxiv.org/abs/1803.02999>
    - ✓ <https://keras.io/examples/vision/reptile/>
  - RegNet: Self-Regulated Network for Image Classification
    - ✓ <https://arxiv.org/abs/2101.00590>
    - ✓ <https://paperswithcode.com/paper/regnet-self-regulated-network-for-image>

# Temas dos seminários

- Computer Vision: Image Segmentation

- U-Net

- ✓ [https://link.springer.com/chapter/10.1007/978-3-319-24574-4\\_28](https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28)

- ✓ <https://www.tensorflow.org/tutorials/images/segmentation?hl=pt-br>

- Boundary-Aware Segmentation Network (BASNet)

- ✓ <https://arxiv.org/abs/2101.04704>

- ✓ [https://keras.io/examples/vision/basnet\\_segmentation/](https://keras.io/examples/vision/basnet_segmentation/)

# Temas dos seminários

- Computer Vision: Object Detection

- YOLOv8

- ✓ <https://blog.roboflow.com/whats-new-in-yolov8/>

- ✓ <https://keras.io/examples/vision/yolov8/>

- YOLOv9

- ✓ <https://arxiv.org/abs/2402.13616>

- ✓ <https://docs.ultralytics.com/pt/models/yolov9/#generalized-efficient-layer-aggregation-network-gelan>

- ✓ <https://github.com/WongKinYiu/yolov9?tab=readme-ov-file>

- YOLO-World: Real-Time Open-Vocabulary Object Detection

- ✓ <https://arxiv.org/abs/2401.17270>

- ✓ <https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/zero-shot-object-detection-with-yolo-world.ipynb>

- ✓ <https://www.youtube.com/watch?v=X7gKBGVz4vs&feature=youtu.be>

# Temas dos seminários

- Computer Vision: Object Detection

- RetinaNet

- ✓ <https://arxiv.org/abs/1708.02002>
    - ✓ <https://keras.io/examples/vision/retinanet/>

- Faster Objects, More Objects (FOMO)

- ✓ <https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/fomo-object-detection-for-constrained-devices>
    - ✓ <https://docs.edgeimpulse.com/docs/tutorials/end-to-end-tutorials/object-detection/detect-objects-using-fomo>

- DETRs Beat YOLOs on Real-time Object Detection

- ✓ <https://arxiv.org/abs/2304.08069>
    - ✓ <https://github.com/lyuwenyu/RT-DETR>

# Temas dos seminários

- Computer Vision: Image Enhancement
  - MIRNet
    - ✓ <https://arxiv.org/abs/2003.06792>
    - ✓ <https://keras.io/examples/vision/mirnet/>
  - Enhanced Deep Residual Networks for Single Image Super-Resolution (EDSR)
    - ✓ <https://arxiv.org/abs/1707.02921>
    - ✓ <https://keras.io/examples/vision/edsr/>
- Computer Vision: Performance recipes
  - Learning to Resize in Computer Vision
    - ✓ <https://arxiv.org/abs/2103.09950v1>
    - ✓ [https://keras.io/examples/vision/learnable\\_resizer/](https://keras.io/examples/vision/learnable_resizer/)

# Temas dos seminários

- Computer Vision: 3D Object Reconstruction
  - TripoSR: Fast 3D Object Reconstruction from a Single Image
    - ✓ <https://arxiv.org/abs/2403.02151>
    - ✓ <https://paperswithcode.com/paper/triposr-fast-3d-object-reconstruction-from-a>

# Temas dos seminários

- Reinforcement Learning

- Deep Q-Learning

- ✓ <https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf>
    - ✓ [https://keras.io/examples/rl/deep\\_q\\_network\\_breakout/](https://keras.io/examples/rl/deep_q_network_breakout/)
    - ✓ [https://www.tensorflow.org/agents/tutorials/0\\_intro\\_rl?hl=pt-br](https://www.tensorflow.org/agents/tutorials/0_intro_rl?hl=pt-br)

- Deep Deterministic Policy Gradient (DDPG)

- ✓ <https://arxiv.org/pdf/1509.02971.pdf>
    - ✓ [https://keras.io/examples/rl/ddpg\\_pendulum/](https://keras.io/examples/rl/ddpg_pendulum/)

- Actor Critic Method

- ✓ <https://inria.hal.science/hal-00840470/document>
    - ✓ [https://keras.io/examples/rl/actor\\_critic\\_cartpole/](https://keras.io/examples/rl/actor_critic_cartpole/)



# Temas dos seminários

- Natural Language Processing: Text summarization
  - Bidirectional Autoregressive Transformer (BART)
    - ✓ <https://arxiv.org/abs/1910.13461>
    - ✓ [https://keras.io/examples/nlp/abstractive\\_summarization\\_with\\_bart/](https://keras.io/examples/nlp/abstractive_summarization_with_bart/)
- Natural Language Processing: Automatic Speech Recognition (ASR)
  - Automatic Speech Recognition with Transformer
    - ✓ <https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
    - ✓ [https://keras.io/examples/audio/transformer\\_asr/](https://keras.io/examples/audio/transformer_asr/)
- Natural Language Processing: Speech Synthesis
  - WaveNet: A Generative Model for Raw Audio
    - ✓ <https://arxiv.org/abs/1609.03499>
    - ✓ <https://paperswithcode.com/paper/wavenet-a-generative-model-for-raw-audio>

# Temas dos seminários

- Graph Neural Networks
  - ✓ <https://arxiv.org/abs/2101.11174>
  - ✓ <https://paperswithcode.com/paper/graph-neural-network-for-traffic-forecasting>
- Spiking Neural Networks
  - ✓ <https://arxiv.org/abs/2109.12894>
  - ✓ <https://analyticsindiamag.com/a-tutorial-on-spiking-neural-networks-for-beginners/>
  - ✓ <https://guillaume-chevalier.com/spiking-neural-network-snn-with-pytorch-where-backpropagation-engenders-stdp-hebbian-learning/>
  - ✓ <https://snntorch.readthedocs.io/en/latest/tutorials/index.html>

# Temas dos seminários

- Adversarial Attacks
  - ✓ <https://arxiv.org/abs/1706.06083>
  - ✓ <https://paperswithcode.com/paper/towards-deep-learning-models-resistant-to>
- Kolmogorov-Arnold Networks (KANs)
  - ✓ <https://arxiv.org/abs/2404.19756>
  - ✓ <https://github.com/KindXiaoming/pykan?tab=readme-ov-file>
- DiffMOT: A Real-time Diffusion-based Multiple Object Tracker with Non-linear Prediction
  - ✓ <https://arxiv.org/abs/2403.02075>
  - ✓ <https://github.com/Kroery/DiffMOT>
- Liquid Time-constant Networks
  - ✓ <https://arxiv.org/abs/2006.04439>
  - ✓ [https://github.com/raminmh/liquid\\_time\\_constant\\_networks](https://github.com/raminmh/liquid_time_constant_networks)

# Temas dos seminários

- Receitas para melhoria do desempenho de modelos
  - Knowledge Distillation
    - ✓ <https://arxiv.org/abs/1503.02531>
    - ✓ [https://keras.io/examples/vision/knowledge\\_distillation/](https://keras.io/examples/vision/knowledge_distillation/)
  - Gradient Centralization
    - ✓ <https://arxiv.org/abs/2004.01461>
    - ✓ [https://keras.io/examples/vision/gradient\\_centralization/](https://keras.io/examples/vision/gradient_centralization/)
  - VeLO: Training Versatile Learned Optimizers by Scaling Up
    - ✓ <https://arxiv.org/pdf/2211.09760>
    - ✓ <https://velo-code.github.io/>

# Referências

- [1] “Tensorflow code examples”, <https://keras.io/examples/>
- [2] “Pytorch code examples”, <https://pytorch.org/examples/>
- [3] “Papers with code”, <https://paperswithcode.com/>
- [4] Sebastian Raschka and Vahid Mirjalili, “Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2”, Packt Publishing, 2019.
- [5] Ivan Vasilev, “Advanced Deep Learning with Python: Design and implement advanced next-generation AI solutions using TensorFlow and PyTorch”, Packt Publishing, 2019.
- [6] Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", 3rd ed., O'Reilly Media, 2022.
- [7] Raschka, Sebastian, et al., “Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python”, Packt Publishing Ltd, 2022.
- [8] C. M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.
- [9] “Livros”, <http://tinyurl.com/tp558-books>

# Avisos

- Todo material do curso está disponível no GitHub:
  - <https://github.com/zz4fap/tp558-adv-ml>
- Como usar o Google Colab
  - [https://www.youtube.com/watch?v=inN8seMm7UI&ab\\_channel=TensorFlow](https://www.youtube.com/watch?v=inN8seMm7UI&ab_channel=TensorFlow)
  - <https://www.youtube.com/watch?v=inN8seMm7UI>
- Python Crash Course
  - <https://www.youtube.com/watch?v=pq4NNIYar9o&list=PLRc6ZYt68prVXAhwY1JD6DFc3BJGmJriq&pp=gAQBiAQB>
- Horário de Atendimento
  - Todas as quartas-feiras das 17:30 às 18:30.
  - Presencialmente ou remotamente.

Perguntas?

Obrigado!