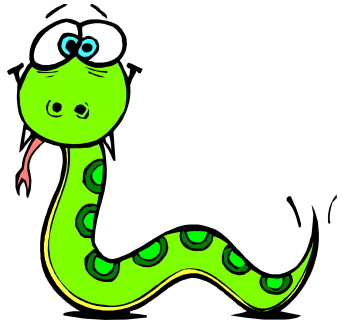
Introduction to Machine Learning in Python



Welcome!!

Instructor: Tugce Gurbuz

July 14th 2022

Who is Tugce?

- First year Ph.D. student in Quantitative Life Sciences department @McGill
- Before McGill -> psychology undergrad @Bilkent University, Turkey

Who is Tugce?

- First year Ph.D. student in Quantitative Life Sciences department @McGill
- Before McGill -> psychology undergrad @Bilkent University, Turkey
- Neuro-AI research <3 <3
  - Perceptual Learning

Who is Tugce?

Let's stay in touch: @tugcdroid

- First year Ph.D. student in Quantitative Life Sciences department @McGill
- Before McGill -> psychology undergrad @Bilkent University, Turkey
- Neuro-AI research <3 <3 -> Perceptual Learning

LEXI IN TECH

https://www.youtube.com/channel/UC1XlzpQpKnfFZP3jGgdM2MQ

Welcome to Introduction to ML in Python!

TEXT DESCRIPTION

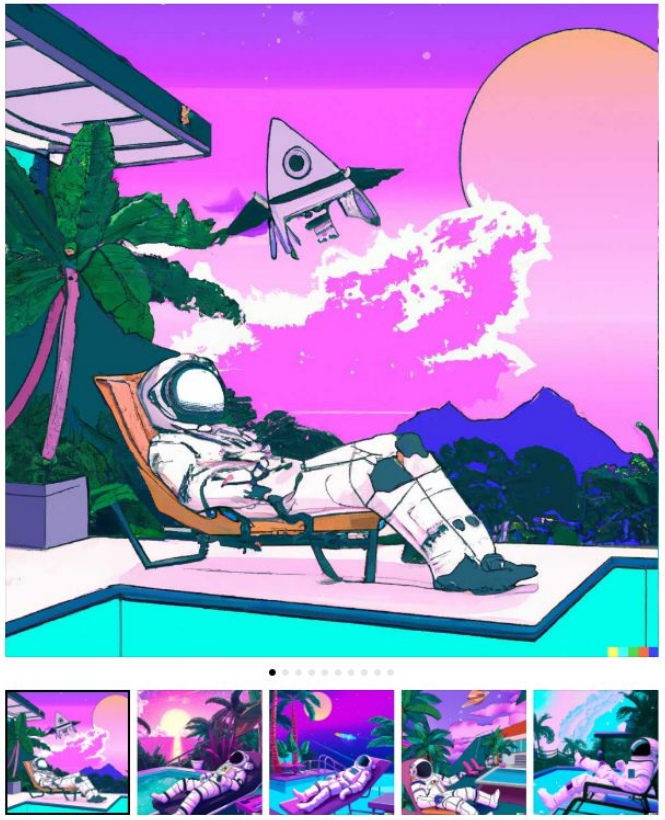
An astronaut Teddy bears A bowl of soup

riding a horse lounging in a tropical resort in space playing basketball with cats in space

in a vaporwave style as pixel art in a photorealistic style

→

DALL·E 2

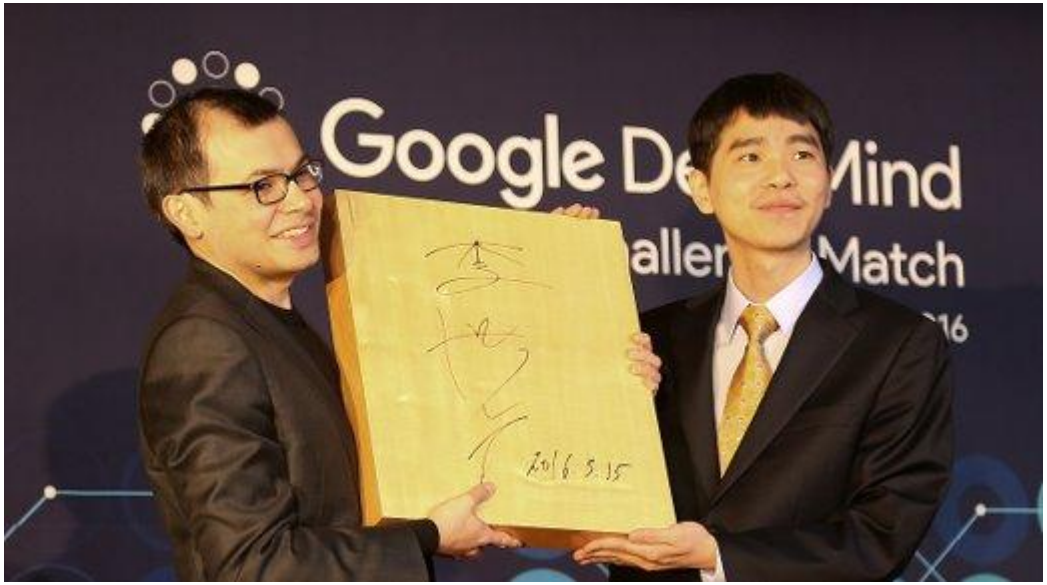


https://openai.com/dall-e-2/#demos

Welcome to Introduction to ML in Python!

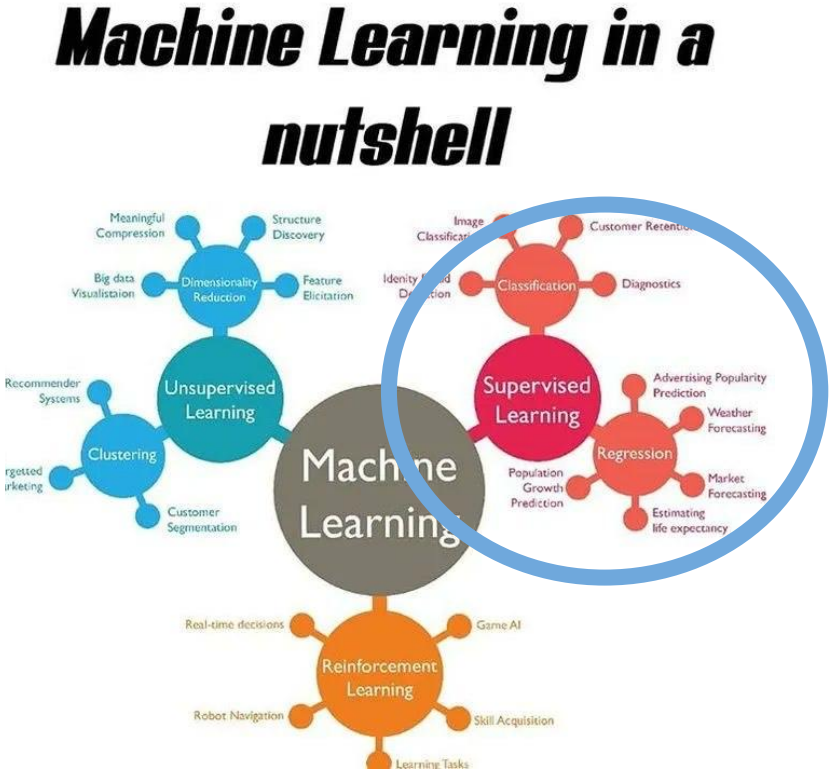
Gatys, Ecker, Bethge, CVPR 2016

Welcome to Introduction to ML in Python!



Welcome to Introduction to ML in Python!

- Which machine learning we will be learning?



Welcome to Introduction to ML in Python!

July 14th: Basics of ML (Module 1)

- Basics and Pytorch

Welcome to Introduction to ML in Python!

July 14th: Basics of ML (Module 1)

- Basics and Pytorch
- Multi Layer Perceptrons

PyTorch

Welcome to Introduction to ML in Python!

July 14th: Basics of ML (Module 1)

- Basics and Pytorch
- Multi Layer Perceptrons
- Optimization

PyTorch

Welcome to Introduction to ML in Python!

July 14th: Basics of ML (Module 1)

- Basics and Pytorch
- Multi Layer Perceptrons
- Optimization
- Regularization

PyTorch

Welcome to Introduction to ML in Python!

July 15th: Convolutional Neural Networks (Module 2)

- Introduction to convolutional neural networks

Welcome to Introduction to ML in Python!

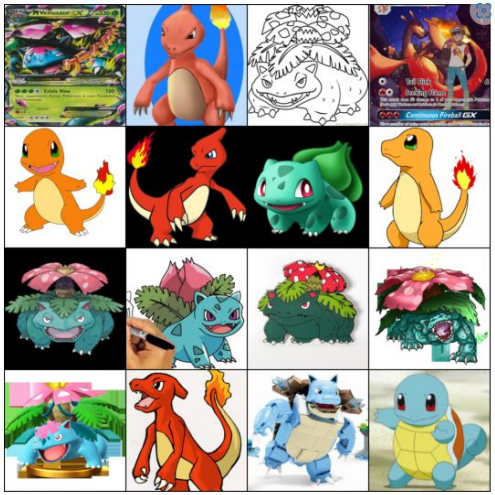
July 15th: Convolutional Neural Networks (Module 2)

- Introduction to convolutional neural networks
- Modern CNNs
  - Transfer Learning

Welcome to Introduction to ML in Python!

July 15th: Convolutional Neural Networks (Module 2)

- Introduction to convolutional neural networks
- Modern CNNs
  - Transfer Learning



Welcome to Introduction to ML in Python!

July 15th: Convolutional Neural Networks (Module 2)

- Introduction to convolutional neural networks
- Modern CNNs
  - Transfer Learning

Welcome to Introduction to ML in Python!

July 15th: Convolutional Neural Networks (Module 2)

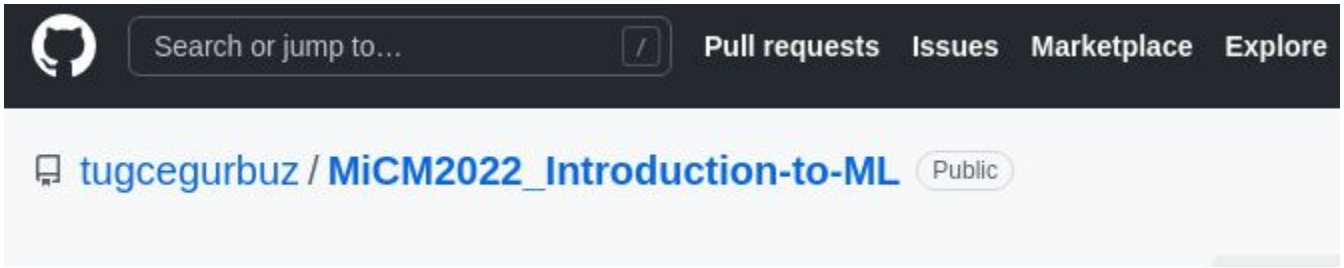
- Introduction to convolutional neural networks
- Modern CNNs
  - Transfer Learning
  - Ethical impacts of AI on the society

Where the lecture materials come from, thank you NMA!

Neuromatch Academy:
Deep Learning

https://deeplearning.neuromatch.io/tutorials/intro.html

Where are the lecture materials?



https://github.com/tugcegurbuz/MiCM2022_Introduction-to-ML

Schedule

1.00 - 1.50 pm -> ML Time!

1.50 - 2.00 pm -> Break

2.00 - 2.50 pm -> ML Time!

2.50 - 3.00 pm -> Break

3.00 - 3.50 pm -> ML Time!

3.50 - 4.00 pm -> Closing discussions

Zoom Logistics to Avoid Chaos

- Everybody is muted on Zoom by default
- **Please wait for question sections to ask questions**

Zoom Logistics to Avoid Chaos

- Everybody is muted on Zoom by default
- **Please wait for question sections to ask questions**
- **To ask question in the question section:**
  - Raise your hand on Zoom using the hand emoji
  - When it is your turn: Unmute yourself / type your question

Zoom Logistics to Avoid Chaos

- Everybody is muted on Zoom by default
- **Please wait for question sections to ask questions**
- **To ask question in the question section:**
  - Raise your hand on Zoom using the hand emoji
  - When it is your turn: Unmute yourself / type your question

<u>*I WON'T CHECK THE CHAT OUTSIDE OF QUESTION SECTIONS***</u>**

Interactive Coding Exercises

```
for i in range(len(hidden_unit_nums)): # Loop over layers and create each one

  ##################################################################
  # Fill in missing code below (...),
  # Then remove or comment the line below to test your function
  raise NotImplementedError("Create MLP Layer")
  ##################################################################

  out_num = hidden_unit_nums[i] # Assign the current layer hidden unit from l
  layer = nn.Linear(..., ...) # Use nn.Linear to define the layer
  in_num = out_num # Assign next layer input using current layer output
  self.mlp.add_module('Linear_%d'%i, layer) # Append layer to the model with

  actv_layer = eval('nn.%s'%actv) # Assign activation function (eval allows u
  self.mlp.add_module('Activation_%d'%i, actv_layer) # Append activation to t

out_layer = nn.Linear(in_num, output_feature_num) # Create final layer
self.mlp.add_module('Output_Linear', out_layer) # Append the final layer
```

Tell us the chosen one

Show code

```
The chosen one:
```