

AI

Neuroverkko

- prosessoriversio
- sudo apt update
- sudo apt upgrade

sudo apt install python3-pip

sudo apt autoremove

sudo apt autoclean

pip3 install tensorflow - - user (tälle käyttäjälle)

Pip python pakettimanageri

- `pip3 install numpy --user`
- `pip3 install pillow --user`
- `touch fashion_mnist_trainer.py`
- Datasetsi fashion : 60 000 kuvaa, 10 000 testiä varten, datat oikeaan muotoon, 10 eri kategorialla kuvia, 28x28 pixeliä, tehdään opetus, tallennetaan verkko, otetaan kuva netistä, muutetaan mv ja kokeillaan nv datasetin ulkopuoliseen kuvaan, kolme tiedostoa, training opetusta varten, kuvankäsittely ja kolmas käytetään opettua verkkoa

- Python3
- `import tensorflow as tf`, ei näytönohjainta, hyppää käyttään prossua
- 2021-01-20 18:46:22.923784: W
tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'libcudart.so.11.0'; dLError: libcudart.so.11.0: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /opt/ros/noetic/lib
- 2021-01-20 18:46:22.923829: I
tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dLError if you do not have a GPU set up on your machine.

Asennusta

- `>>> print(tf.__version__)` => 2.4.0, tarkista !
- `msg = tf.constant("Hello World")`
- `tf.print(msg)` => Hello World, toimivuuden tarkistus
- `pip3 install numpy --user`, numeerinen laskenta, matriisilaskent
- `pip3 install pillow --user`, Pillow kuvankäsittelyyn
-
-

```
rosp@ubuntu: ~/tf_harjoitus$ ./fashion_mnist_trainer.py
bash: ./fashion_mnist_trainer.py: /user/bin/python3: bad interpreter: No such file or directory
rosp@ubuntu:~/tf_harjoitus$ ./fashion_mnist_trainer.py
2021-01-19 07:40:41.799997: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /opt/ros/noetic/lib
2021-01-19 07:40:41.800067: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
32768/29515 [=====] - 0s 2us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26427392/26421880 [=====] - 5s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
6192/5140 [=====] - 0s 2us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4423680/4422102 [=====] - 1s 0us/step
(60000, 28, 28)
rosp@ubuntu:~/tf_harjoitus$
```

```
fashion_mnist_trainer.py  jeans.jpg  saved_model  
asp@ubuntu:~/tf_harjoltus$ chmod +x fashion_mnist_predictor.py  
asp@ubuntu:~/tf_harjoltus$ ./fashion_mnist_predictor.py
```

Enter img path:

neuroverkko

```
u)? /proc/driver/nvidia/version does not exist
2021-01-20 23:52:20.623078: I tensorflow/compiler/jit/xla_gpu_device.
Model: "sequential"
```

Layer (type)	Output Shape	Param #
=====	=====	=====
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 128)	100480
dense_1 (Dense)	(None, 10)	1290
=====	=====	=====
Total params: 101,770		
Trainable params: 101,770		

```
rosp@ubuntu:~/tf_harjoitus$ ./fashion_mnist_trainer.py
bash: ./fashion_mnist_trainer.py: /user/bin/python3: bad interpreter: No such file or directory
rosp@ubuntu:~/tf_harjoitus$ ./fashion_mnist_trainer.py
2021-01-19 07:40:41.799997: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'libcudart.so.11.0'; dlderror: libcudart.so.11.0: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /opt/ros/noetic/lib
2021-01-19 07:40:41.800067: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
32768/29515 [=====] - 0s 2us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26427392/26421880 [=====] - 5s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
8192/5148 [=====] - 0s 2us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4423680/4422102 [=====] - 1s 0us/step
(60000, 28, 28)
rosp@ubuntu:~/tf_harjoitus$
```


- Relu , niin kauna kuin nolla ei aktivointia ja sen jälkeen lineaaristi aktivointi
- Sigmoid ei voi koskaan mennä yli yhden, laskennallisesti raskaampi

fashion_mnist_trainer.py X

fashion_mnist_trainer.py > ...

```
33
34 train_images = train_images / 255.0
35 test_images = test_images / 255.0
36
37 model = tf.keras.Sequential([
38     tf.keras.layers.Flatten(input_shape=(28,28)),
39     tf.keras.layers.Dense(128, activation='relu'),
40     tf.keras.layers.Dense(10)
41 ])
42
43 model.compile(optimizer='adam',
44               loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
45               metrics=['accuracy'])
46
47 model.fit(train_images, train_labels, epochs=10)
48
49 test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=2)
50
51 print('\nTest accuracy:', test_acc)
52
53 model.save('saved_model')
54
55
```

- "https://pixy.org/src/154/
thumbs350/1547777.jpg"

```
rosp@ubuntu: ~/tf_harjotus
training.py:1461 run_step **
    outputs = model.predict_step(data)
    /home/rosp/.local/lib/python3.8/site-packages/tensorflow/python/keras/engine
training.py:1434 predict_step
    return self(x, training=False)
    /home/rosp/.local/lib/python3.8/site-packages/tensorflow/python/keras/engine
base_layer.py:1012 __call__
    outputs = call_fn(inputs, *args, **kwargs)
    /home/rosp/.local/lib/python3.8/site-packages/tensorflow/python/keras/engine
sequential.py:375 call
    return super(Sequential, self).call(inputs, training=training, mask=mask

    /home/rosp/.local/lib/python3.8/site-packages/tensorflow/python/keras/engine
functional.py:424 call
    return self._run_internal_graph(
    /home/rosp/.local/lib/python3.8/site-packages/tensorflow/python/keras/engine
functional.py:560 _run_internal_graph
    outputs = node.layer(*args, **kwargs)
    /home/rosp/.local/lib/python3.8/site-packages/tensorflow/python/keras/engine
base_layer.py:998 __call__
    input_spec.assert_input_compatibility(self.input_spec, inputs, self.name

    /home/rosp/.local/lib/python3.8/site-packages/tensorflow/python/keras/engine
input_spec.py:255 assert_input_compatibility
    raise ValueError(

ValueError: Input 0 of layer dense is incompatible with the layer: expected
axis -1 of input shape to have value 784 but received input with shape (None, 28
)
```

- `pip3 uninstall tensorflow --user`
-
- `jiiteepee@ubuntu:~/tf_harjoitus$./img_download_preprocessing.py`
- `[[255 255 255 ... 255 255 255]`
- `[255 255 255 ... 255 255 255]`
- `[255 255 255 ... 255 255 255]`
- `...`
- `[255 255 255 ... 255 255 255]`
- `[255 255 255 ... 255 255 255]`
- `[255 255 255 ... 255 255 255]]`
- `<PIL.Image.Image image mode=L size=28x28 at 0x7FBD17D964F0>`

model.summary()

- Model: "sequential"

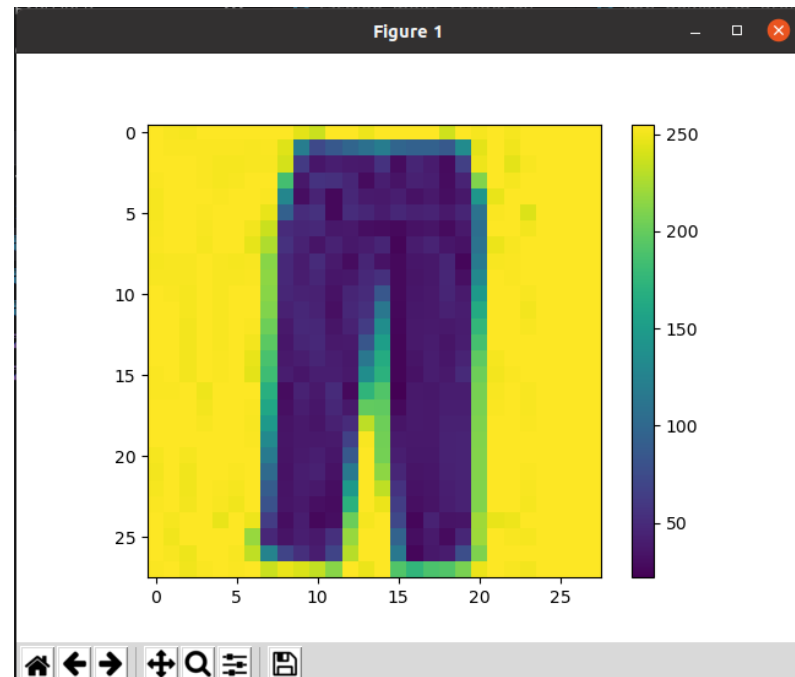
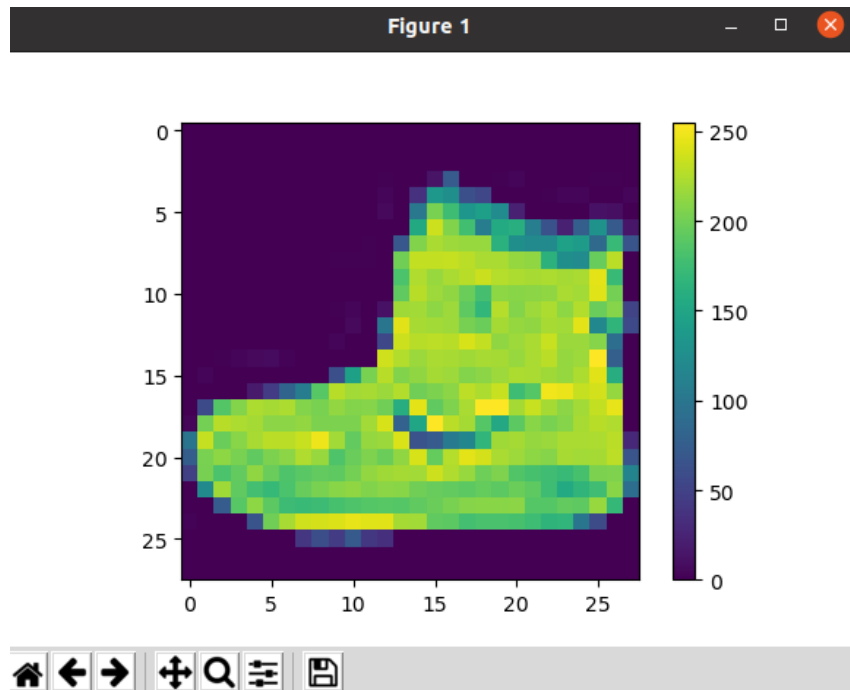
•	<hr/>	
•	Layer (type)	Output Shape
•		Param #
•	<hr/>	
•	flatten (Flatten)	(None, 784)
•		0
•	<hr/>	
•	dense (Dense)	(None, 128)
•		100480
•	<hr/>	
•	dense_1 (Dense)	(None, 10)
•		1290
•	<hr/>	
•	Total params: 101,770	
•	Trainable params: 101,770	
•	Non-trainable params: 0	

```
print(prediction_single)
```

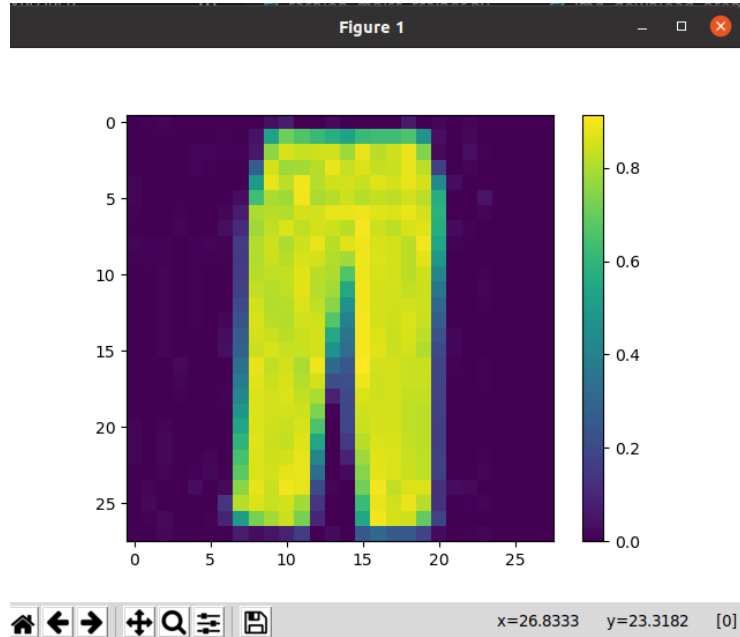
- 2021-01-21 12:58:48.719021: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enabled (registered 2)
- 2021-01-21 12:58:48.719530: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency: 2711995000 Hz
- [[-2.6885443 -39.276993 -5.719284 -30.029842 -25.050695 -
21.868929
- 21.17138 -34.365486 34.22429 -43.133358]]

print(prediction_single ja single2)
löytää luokasta Trousers

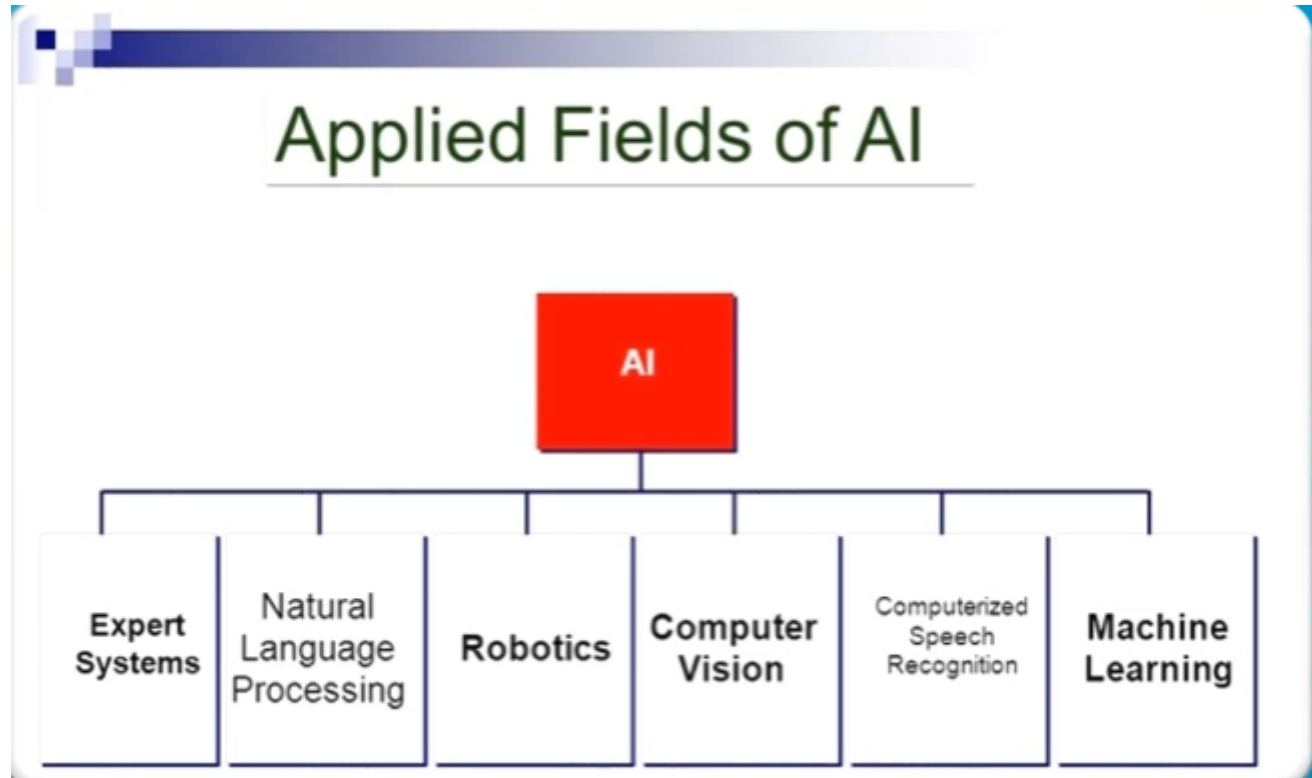
- 2021-01-21 13:35:31.979652: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency: 2711995000 Hz
- [[-2.6885443 -39.276993 -5.719284 -30.029842 -25.050695 -
21.868929
- 21.17138 -34.365486 34.22429 -43.133358]]
- [[-2.115005 7.9907064 -5.425074 -3.1688159 -2.1182737 -
14.250074
- -1.3722041 -20.571974 -7.6881385 -19.027512]]



```
np_image_inverted = 255 - np_image  
np_image_normalized_inverted = np_image_inverted / 255.0
```



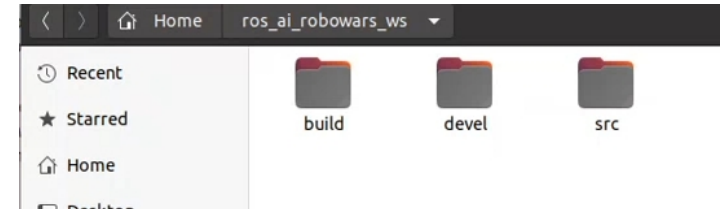
Perinteinen jaottelu



- Expert, säännöt tarkkaan määritelty, rajallinen ongelma, shakki, deep blue -96
- Natural language processing, ennalta määrättyt vastaukset ,chatbot, kielten käännökset
- Robotics, perinteiset, kalman, liikkeenohjaus,slam, lokalisointi eivät käytä koneoppimista
- Computer vision, ihmisen määrittämät, opetus,features, suurin osa on tällaisia koska olosuhteet vakiot, valaistus !! laskenta kamerassa, ei vie cpu tehoa
- Computer speech recognition, kohtuudella, murteet , puhetyyli!
- Machine Learning, deep learning, neuroverkoissa useita kerroksia, nopeet PC:t, näytönohjain tekee rinnakkaisen laskennan, CPU laskee helposti peräjälkeen, back propagation algoritmi, tunnetusta virhe takaisinpäin
-
-

Robowars, kolmas luento

- https://github.com/tonaalt/ros_ai_robowars_ws.git
- `jiitepee@ubuntu:~$ git clone https://github.com/tonaalt/ros_ai_robowars_ws.git`, terminaalissa
- Githubista haetaan catkin työtila, src kansiossa viisi pakettia, 2 eri robottia, cazebo conttrlolleer, pari maililamaa, eri versioita areenasta
- `/ros_ai_robowars_ws/src$` , pelkkä source pitää olla vain latauksen jälkeen
- `cd ros_ai_robowars_ws`, `catkin_make` työtilan juuressa vain source, build ja develi pois, ei toimi eri koneilla, jos hajoaa niin poista build ja devel, `catkin_make clean` toimii myös, prosessi alusta
- Tehdään catkin make, kääntää c++ paketit, tekee työtilan
- Työtilan juuressa, `rosdep install --from-paths ./src --ignore-packages-from-source --rosdistro noetic -y`
- Katsoo mitä kaikkea työtilan kääntäminen tarvitsee, mitä ohjelmia ei ole, riippuvuudet, edellinen komento, resolving dependencies
- `jiitepee@ubuntu:~/ros_ai_robowars_ws$`, c, code .
-
- Robotti1, urdf määrittää geometrian, yleensä 3D mallia, STL-muoto
- Hitbox ja ulkonäkö erilaisia, helpottaa laskentaa

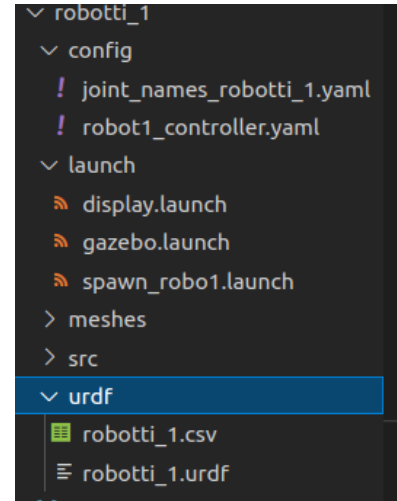


Robowars

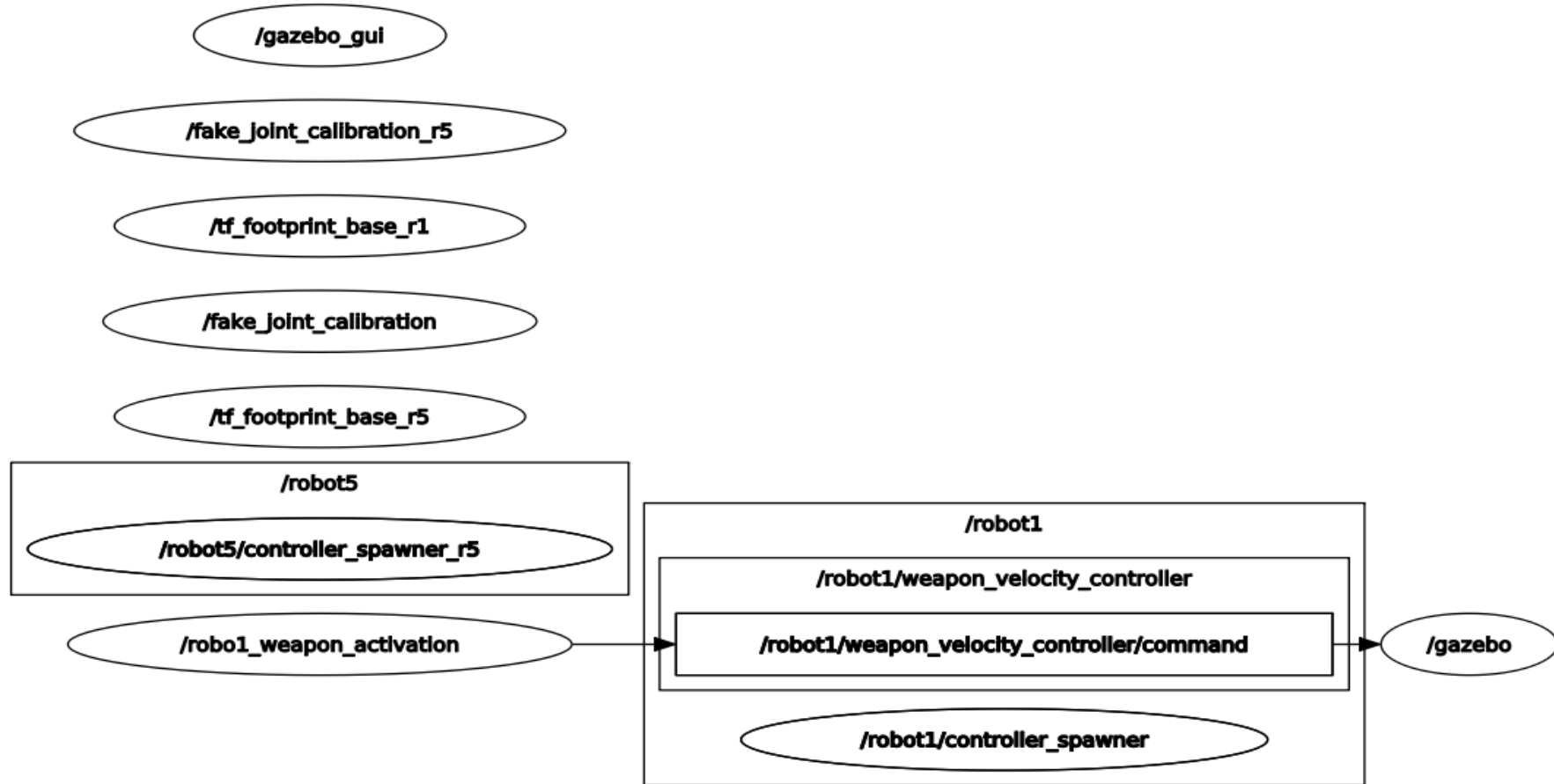
- Src kansio, ros_control paketit ei tarvite puuttua,robotti1, urdf kansiossa , robotin geometria, 3d malli, stl-muotona, collison ja ulkonäkö erilaiset, hitbox yksinkertaistettu muoto, lauch tiedosto. 16:59
- Miltä robotti näyttää,jiiteep@ubuntu:~/ros_ai_robowars_ws/src/robotti_1/launch\$, roslaunch robotti_1 display.launch, ei toiminut
- ERROR: cannot launch node of type [robot_state_publisher/state_publisher]: Cannot locate node of type [state_publisher] in package [robot_state_publisher]. Make sure file exists in package path and permission is set to executable (chmod +x)
- jiiteep@ubuntu:~/ros_ai_robowars_ws/src/robotti_1/launch\$ roslaunch robotti_1 display.launch
- roslaunch robotti_1 gazebo.launch, ei toiminut
- jiiteep@ubuntu:~/ros_ai_robowars_ws/src/robotti_1/launch\$ roslaunch robotti_1 gazebo.launch
- roslaunch samk_robowar_world samk_robowar_arena.launch
- jiiteep@ubuntu:~/ros_ai_robowars_ws/src/robotti_1/launch\$ roslaunch samk_robowar_world samk_robowar_arena.launch
- samk_roboworl_word, package xml, täältä voi kommata alemman käyttöön jos ei toimi
- Hakee automaattisesti ~/.gazebo/models, arena.world, laita tänne jos mallit ei löydy
- Voi lisätä käsin Gazebo path polkuun, path pitää muuttaa jos kone vaihtuu
- Odometry löytyy tarkka paikka, oikealta robotilta, robotti 1 löytyy samk
- Miten ultran perusteella voidaan laskea robotin paikka, vaikea tehdä laskennallisesti ja jos antureiden paikat muuttuvat, kaksi reikää sivulla
- Liikutetaan molempia satunnaisesti ja kerätää ultran tieto ja kerätään ground truth paikka simulaatiossa ja sen jälkeen opetetaan mitkä sensoridatat vastaavaat mitätkin paikkaa, koodi tehdään vain kerran, koodi helppo ylläpitää
- Periaate ei muutu vaikka mitattaisiin värähtelyjä sensoridataa, kierrosnopeuden mittausta, samat periaatteet toimivat
- 1.Ensin kerätään data, (rosbag muodossa, kylläkin hidasta, opetus kuitenkin tunteja, voidaan opettaa tämän perusteella) , 2: rosbagiin ja kirjoitetaan parser joka kirjoittaa ja tuuppaa csv muotoon. Jos refor informant, niin rosbag toimii
- Kirjoitetaan rosnod joka kirjoittaa suoraan csv;hen, kun csv on valmis , tehdään neuroverkko, ajetaan , lopuksi tehdään node joka lukee sensoridatat
- Köyttää tensorflowta ja opetettu verkko, reaaliaikaisesti koittaa laskea omaa paikkaansa, voidaan tarkkailla ja validoida oikean paikan ja lasketun paikan ero.

- Rostopic list, /robot1/odom, inertiamittaus
- Weapon velocity, terän nopeus
- Robotti1 2 rengasta, 5 neljä rengasta + lidar

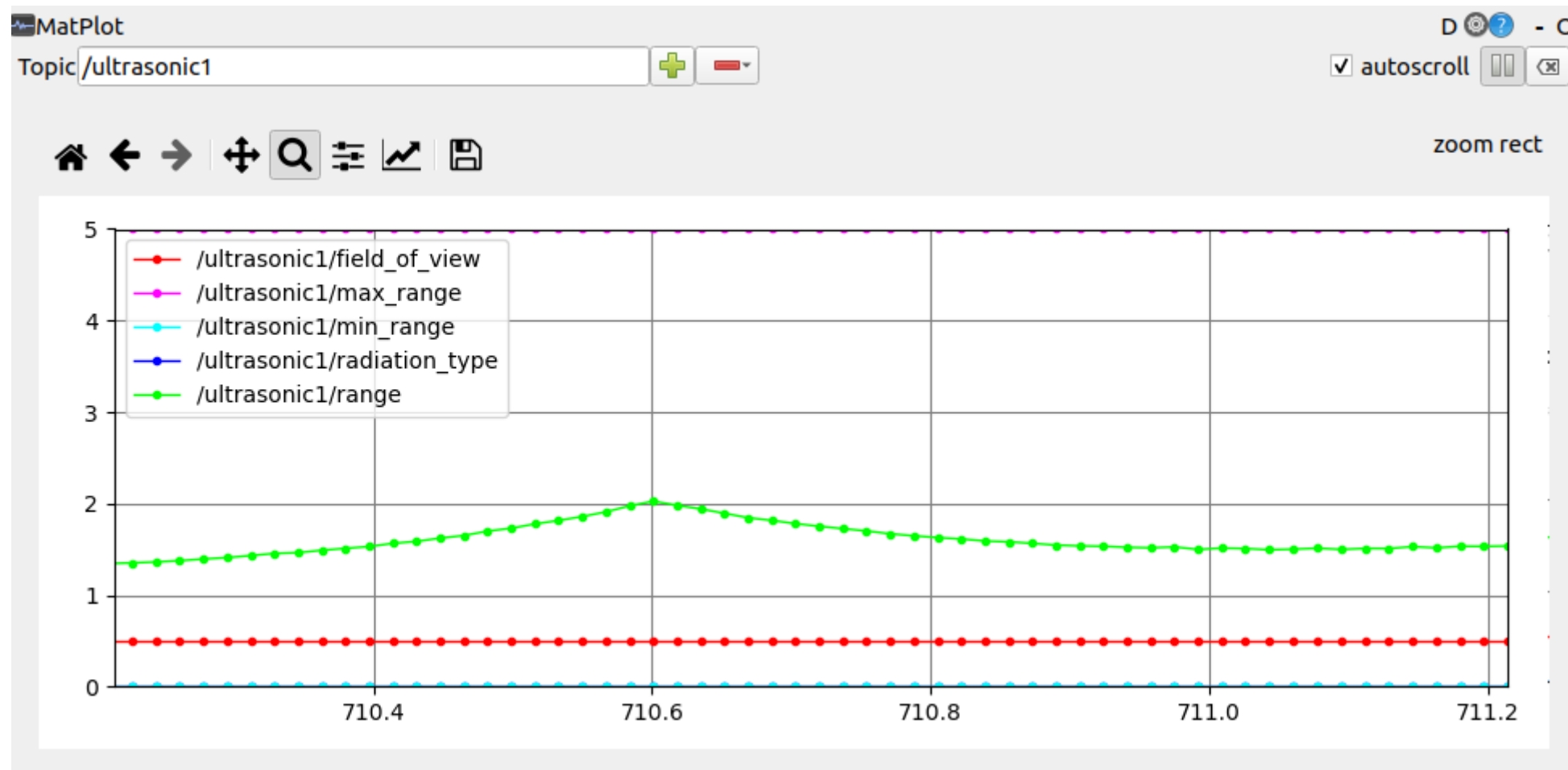
- Urdf robotin rakenne, controllerit ajemista varten, kamera
- Kamerakuvat: `roslaunch rqt_image_view rqt_image_view`, valikosta mitä topiccia halutaan katsoa
- Liikutetaan robottia käsin, `rostopic pub -r 10 (r antaa 10/s ohjetta)`
- `/robot1/wheel_l_velocity_controller/command std_msgs/Float64 "data: 0.0"`, tabbi autocomplitta
- Yksi 64 bittinen luku, 6.0, `rostopic pub -r 10 /robot1/wheel_l_velocity_controller/command`
- `std_msgs/Float64 "data: 6.0"`
-
- `rostopic echo /ultrasonic1`, range vaihtelee, menee lähelle nolla
- `rostopic info /ultrasonic1`, Type: `sensor_msgs/Range`, timestampit mukana, /gazebo (<http://ubuntu:34013/>, data yksi numero)
- Robot_1. Urdf , visualize , true
- Sensorien asetus, robotti1 , urdf , kameralla määrittäviä, laajakulma, rgb, 320x320 resoluutio, kuinka lähelle ja kauas, vääristymät, UA:sensorit visualize : True, 6 kpl, voi määrittellä kuinka usien data näkyy, rate60, kuinka kapea/laaja keila hori/verti, pystytään säätämään mikä on on sensorien oikea vaste, range, mihin mini/ja maks näkee
- Nested launch file, päälaunch `roslaunch samk_robo_arena`, käynnistetään robotti1 ja robotti5, päälaunch jakaa osalaunchien käynnistykset,yaml
- robotti1/config , YAML , tiedosto,määrittää controllerit , state:tila ulospäin, vasen ja oikea rengas ja ase, annettu random pid arvot
- Spawn rob launch : spawnataan , annetaan namespace, annetaan mihin spawnataan , face calibration, ladataan controllerit, voidaan suoraan roslaunch rostopic käskyjä viimeisenä, ase lähtee pyörimään heti käynnistettäessä
-
- Launchissa voidaa antaa rostopic käskyjä, paketti ,tyyppi ja joku nimi argumentti ja publish ja minne



roslaunch rqt_graph rqt_graph



roslaunch rqt_plot rqt_plot



- Miten kerätään data

- , tarvitaan ultrasonicit, odometria missä ollaan, ground truth
- `jiiteepee@ubuntu:~/ros_ai_robowars_ws/src$`
- `catkin_create_pkg robot_position rospy nav_msgs sensor_msgs std_msgs message_generation message_runtime roscpp`
- Kääntyykö `catkin_make`, `source devel/setup.bash`
- Gazebo simulation dokumentation
- `jiiteepee@ubuntu:~/ros_ai_robowars_ws/src/robot_position$ ls`
- `src write_to_csv.py`
- `jiiteepee@ubuntu:~/ros_ai_robowars_ws/src/robot_position$ ls`
- `src write_to_csv.py`
- Toiseen terminaliin: `jiiteepee@ubuntu:~/ros_ai_robowars_ws$ source devel/setup.bash`
- `roslaunch samk_robowar_world samk_arena_no_pit.launch`
- `Rosrun robot_position write_to_csv`
-

- Rosrun rqt_image_view rqt_image_view
- rostopic pub -r 10 /robot1/wheel_l_velocity_controller/command std_msgs /float64 "data :6.0"
- Rostopic echo ultrasonic_1
- Rostopic info/ultrasonic1
- Robotti1 . Urdf, kamera
- <visualize>false</visualize> , tämä Trueksi

```
... shutting down processing monitor complete
done
rosp@ubuntu:~/ros_at_robrowars_ws/src/robotti_1/launch$ roslaunch samk_robrowar_
world_samk_robrowar_arena.launch
<architecture independe
</export>
Do you want to install the recommended extensions for
```

- `catkin_create_pkg robot_position rospy nav_msgs sensor_msgs std_msgs message_generation message_runtime roscpp`
- `Rosrun robot_position write_to_csv.py`
- Simulaatio `roslaunch, samk...`, `roslaunch robot_positions, rostopic publisher`
- `rostopic pub -r 10 /robot1/wheel_r_velocity_controller/command std_msgs/Float64 "data: 4.0"`
- `catkin_create_pkg robot_position rospy nav_msgs sensor_msgs std_msgs message_generation message_runtime roscpp`
-

```
106 def go_to_pose_goal(self):
107
108     self.group.set_pose_reference_frame('camera_rgb_optical_frame')
109     self.group.set_end_effector_link('tool0')
110     self.group.clear_pose_targets
111     goal_coord = self.get_pose_coord()
112
113     q = tf.transformations.quaternion_from_euler(-0.30, 0, 0)
114
115     pose_goal = geometry_msgs.msg.Pose()
116     pose_goal.orientation.x = q[0]
117     pose_goal.orientation.y = q[1]
118     pose_goal.orientation.z = q[2]
119     pose_goal.orientation.w = q[3]
120     pose_goal.position.x = goal_coord[0]
121     pose_goal.position.y = goal_coord[1]
122     pose_goal.position.z = goal_coord[2]
123     self.group.set_pose_target(pose_goal)
124
125     plan = self.group.go(wait=True)
126     self.group.stop()
127     self.group.clear_pose_targets()
```

- MoveGroupCommander-luokan `set_pose_target()`-funktio vaatii ohjelmapisteen x-, y-, ja z -sijainti-koordinaatit sekä x-, y-, z-, ja w -orientaatiomuuttujat kvaterniomuodossa. Liikekäskylle on tehtävä orientaatiomuunnos, sillä käytetty koordinaatisto poikkeaa työalueen pinnan normaalista. Koska kvater-nioiden käsin laskenta on huomattavan vaativaa, määritellään orientaatiomuuttujat tf-kirjaston `transformations`-luokan `quaternion_from_euler()`-funktion avulla, jolla suoritetaan muunnos radiaaneista kva-terniomuotoon. Alkiot tallennetaan listaan q, joista niitä voidaan käyttää orientaation määrittelyssä. Lii-kekäskyn suunnitteluun kutsutaan `geometry_msgs.msg.Pose`-viestityyppiä ja syötetään q-ja `goal_coord-`
- 26alkiotlistaan. Lopuksi liikekäsky tallennetaan ja suoritetaan sekä poistetaan muistista. Samalla periaat-teella suoritetaan työkappaleen poiminta funktiolla `grasp_object()`(LIITE 1)

Tehtävä 3

- `rostopic pub -r 10 /robot5/wheel_bl_velocity_controller/command std_msgs/Float64 "data: 6.0`
- `rostopic pub -r 10 /robot5/wheel_br_velocity_controller/command std_msgs/Float64 "data: -6.0`
- `rostopic pub -r 10 /robot5/wheel_fl_velocity_controller/command std_msgs/Float64 "data: 6.0`
- `rostopic pub -r 10 /robot5/wheel_fr_velocity_controller/command std_msgs/Float64 "data: 6.0`
-

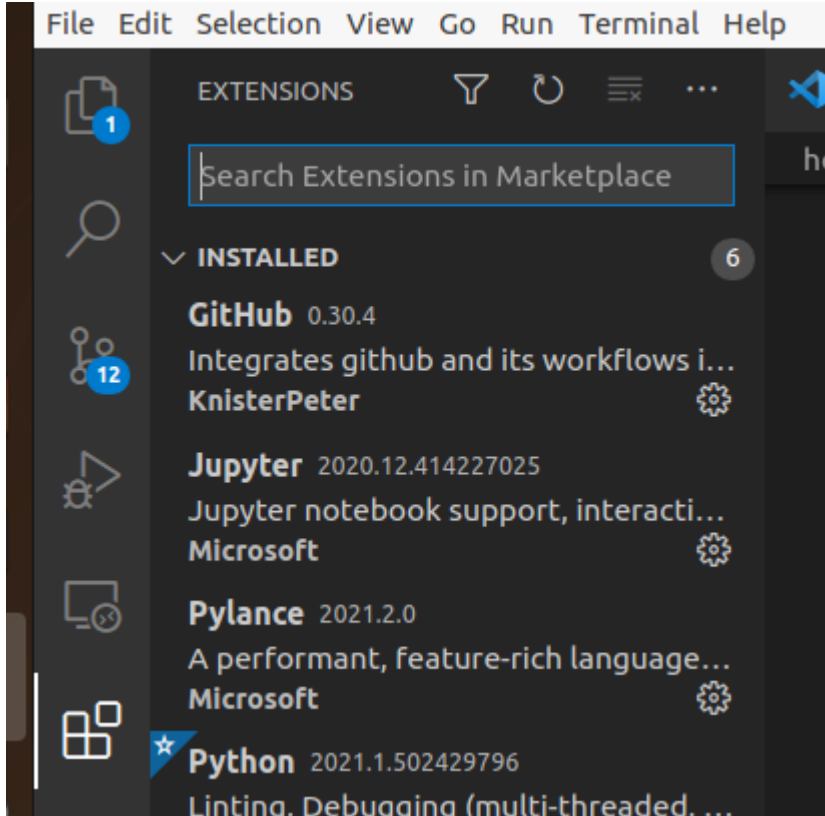
Robotti 1

- Haettiin github robotti, models, models kansioon, kaksi robottia areenalla, node tallensi ultraäänii , positiot ja opetusta varten x ja y paikat
- Tallennettuna csv tiedosto, tehdään opetus, opetusvaihe ei liittyy rossiin vielä, tehtiin paketti robot position, uusi tiedosto src kansioon
- Vasta käytettäessä tule ros mukaan

- Uusi tiedosto, robot_position/src
- Ping 8.8.8.8, testi google dns serverille
- killall gzclient && killall gzserver && killall rosmaster
- rostopic pub -r 10 /robot1/wheel_r_velocity_controller/command std_msgs/Float64 "data: 4.0"
- source ~/ros_ai_robowars_ws/devel/setup.bash,absoluuttinen
- Miksi rosrun rqt_plot rqt_plot ei näytä /robot1/imu tietoa ?

- Haluttiin kerätä data talteen write. To csv , tegtiin data kerralla , avattiin tiedosto, kirjoitetaan csv writerilla, joka kerta kun saadaan data , resetoidaan, muutetaan quaternion euler muotoon normalisoitiin radiaaneiksi, otettiin abs groun tyrut tiedot x ja x , t atallentt, joka 100 000 resetoitiin robottien paikat, comma separated values, yhdell' rivill' on yksi kierros yh det lukemat, 500 arvoa päästiin jonkinnäkiseen tarkkuuteen 10000 riviä on hyvä tarkkuuus,
- Tarkkuutta vois parantaa roll ja pitch tiedot, train position kohdassa , testataan että pipeline toimii, tarin , validation ja data samassa, ei hyvä juttu, jos kunnolla niin kolme erillistä tiedosta
- Käytettiin pandasia, jolla luettiin, datasta nypättin popilla x ja y arvot
- Tehtiin kaikille klemille datasetille , yaw ja kuusi sensoridata, verrattin x ja y
- Yksinkertainen neuroverkko, relu käyttöössä, täysin kytketti, input shape
- Ulostulo layer 2, sissääntulo inputtien leveys määrät pitää olla sam
- Valittiin optimizer, vähän fiksumpi, määriteltiin kuinka nopeasti koittaa oppia, saatta yppiä jos liina nopea opetus, voi hyppiä , häviöfunktio, yritetään minimoida, mse, palautettioon model ja model .fit käynnistettiin opetus, erilliset datat , epokkien määrä 10 , kuinka monta kertaa näkee, pyöritettiin 50 kertaa, jos overfittaa helposti ,lopuksi aina jos löytyy parempi niin tallennettiin, kansioon saved model
- Valmis verkko käyttöön, ei saa tippua kuoppaan, samnalaine kuin csv kirjoitus, keräättiin data , tehtiin model.predict, ladattu valmis model, käy modelilla yhden data ja palauttaa kaksi arvoa, muutettiin stringeiksi, printatiin ground truth arvot, tegtiin oma tpois johon predistced pose julkaistaan, rostopisc echo predictec values
- Seuraavaksi joloa, objektin tunnistusta, kamera gazeboon ja tunnistetaan, pystytään testaamaan koko jono , hidas nopeus, ei 30 kuvaa /s, tarvittaisiin näytönohjain jos nopeempaa tarvetta
- Write to csv kolme erillistä tiedosto, pysäyttää kerään tiedot 70 % koko data 20, 10 % test

github



`sudo apt install git`

Code .

`git config --global user.name " xx "`

`git config --global user.email " xx "`

Tentti AI3

.csv tiedoston nimi tulee sanoista comma separated values

Valitse yksi:

- ☒ Tosi
☐ Epätosi

kun käytät ROSSin ApproximateTimeSynchronizer:i parametrilla määrittää.

Valitse yksi:

- ☐ a. Maksimi arvoa mitä yksittäiseltä sensorilta vu
☐ b. Taukoa, kahden callbackin välillä
☒ c. Kuinka kauan maksimissaan viestiä odotetaan
☐ d. Kuinka usein viesteihin vastataan

Mikä on '/gazebo/reset_simulation' palvelun tyyppi

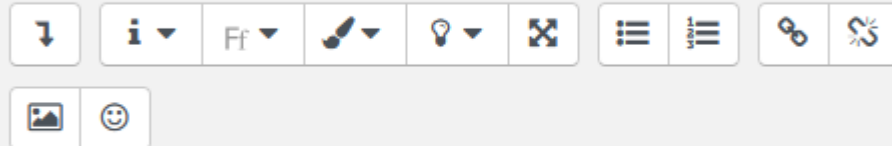
Vastaus: `from std_srvs.srv import Empty`

nav_msgs.msg Odometry viestissä orientaatio on esitetty euler kulmina.

Valitse yksi:

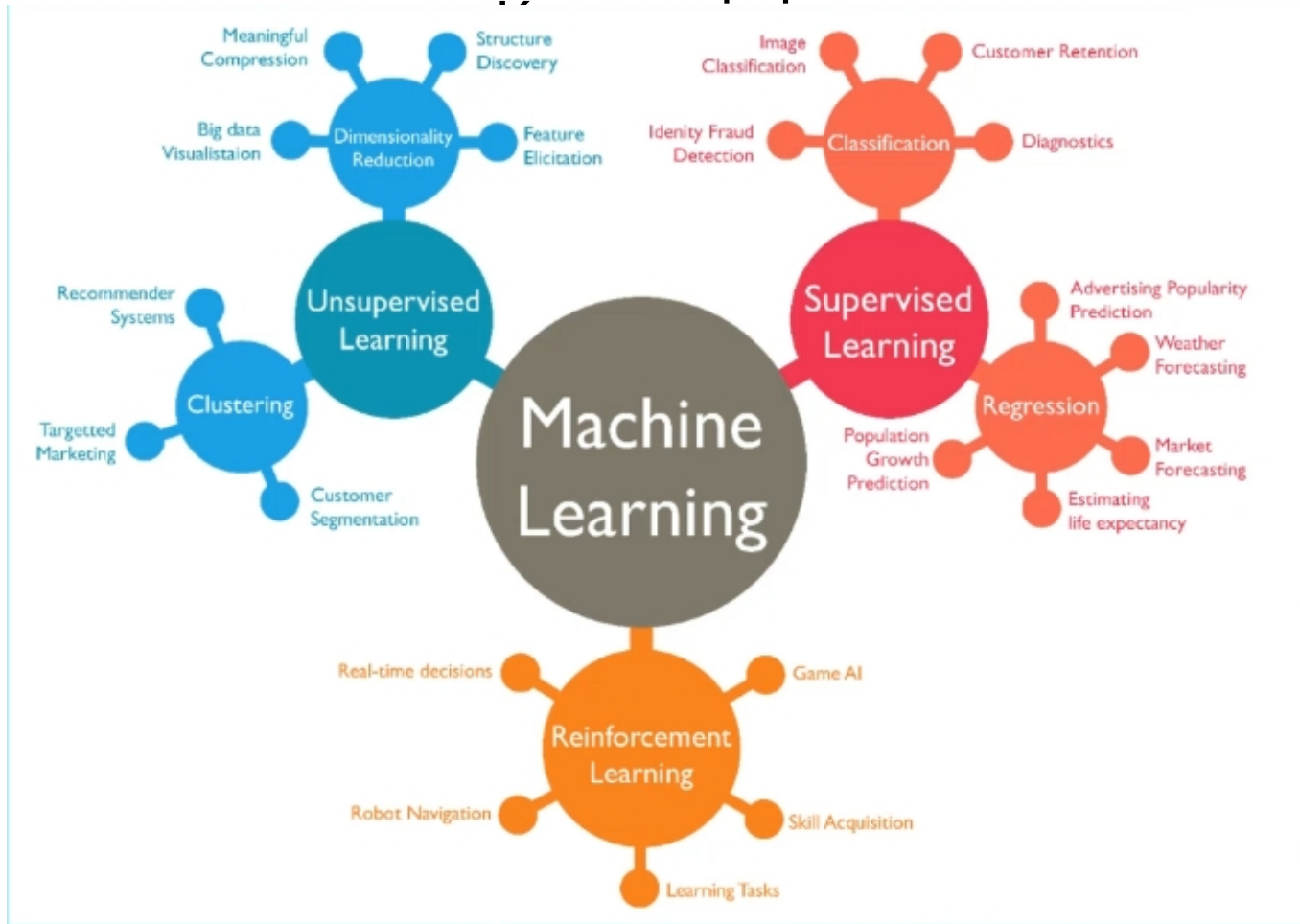
- ☐ Tosi

☒ Miksi tunnilla tehdyssä esimerkissä tallennettiin myös robotin oikea paikka, sensori tietojen lisäksi?



opetettu verkko,neuroverkon avulla voidaan validoida oikean paikan ja lasketun paikan ero "reaaliaikaisesti"

opetetaan mikä sensoridata vastaa mitälin paikkaa



Koneoppiminen, Machine Learning

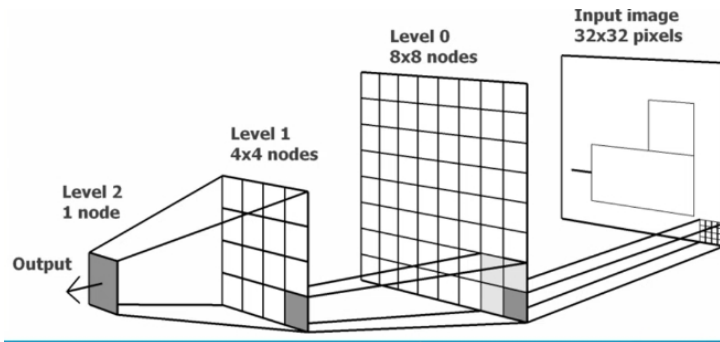
- Joustavampaa ja halvempaa, ei vaadi kuin laskentatehoa
- Useampi taso 14×14 muuttujaa , jokaisella arvo, liittyneenä eteen ja taakse, muuttujapaketti matriisi, opetetaan pikselien liitynnät toisiinsa
- Neuroverkko pyörii pc:ssä, opetus vie tehoa
- Raspberry , usb neural stick, jetson nano, arm, nvidia, tensor flow light, arduinossa pyörii

Supervised Learning

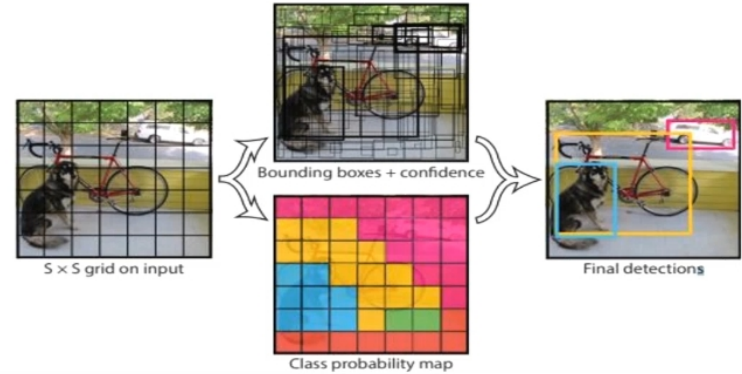
- Ohjattu oppiminen, data on “labelled” eli oikea vastaus on tiedossa, yleisin ja ymmärretty
- Kuvien luokittelu, objektien tunnistus, kuvan segmentointi, regressiomallit
- Kuvien luokittelu:, kissa vai koira , 1 tai 0, neuroverkoissa tarvitaan dataa kuvan syöttämiseen, opetusdatan laatu merkittävä
- Kaksi kansiota kissa ja koira, jaetaan 3 opetus(epoch), validointi, mitä neuroverkko on oppinut ok vai ei, tietty prosentti oikein , testaus, tarkistetaan että löytyy oikea ratkaisu
- Data pitää olla erillään ! Kolme datasettiä, ei saa ajaa liian kauan

Konvoluutio neuroverkko

- 32x32 pixels eka layer, siitä lasketaan 8x8 , 4x4...ja lopputulos 1 tai 0 (mv kuva) , käydään läpi, lasketaan tulos yhdelle pixelille, pyöristetään ykkösen tai nollaan, supistetaan

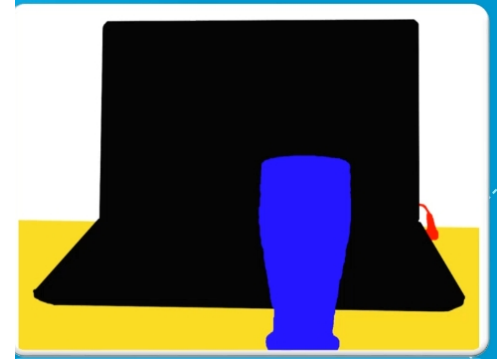


Objektien tunnistus



- Etsitään kuvasta tunnistettavat objektit, ja ympärille "bounding box" sekä määritellään % kuinka varma tunnistus on
- Vaatii laskentatehoa, voidaan aloittaa jonkun opettamasta neuroverkosta, 20-30 kuvaa sekunnissa (fps), mv pudottaa 1/3 tehontarpeen, 320x240 resoluutio riittänee yo. Pienet kuvat tarvii tehoo
- Jos ihminen näkee kuvasta, niin samaan pystyy kone

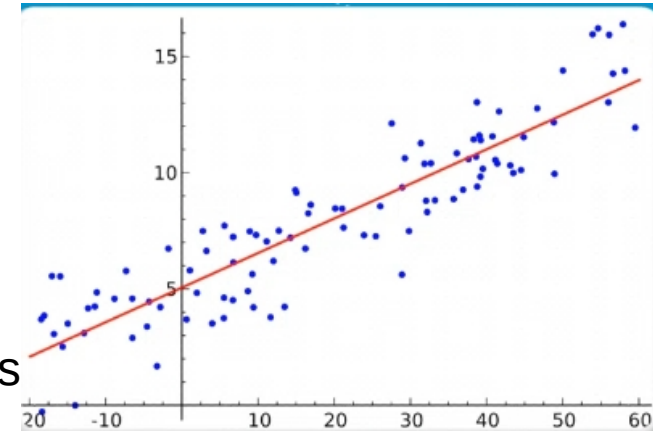
Kuvan segmentointi



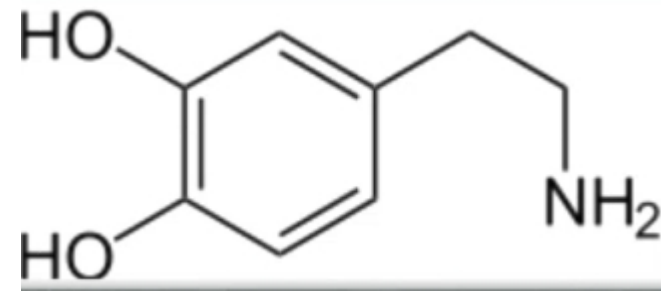
- Erottelee objektit ja niiden sijainnit, turvallisuus on tulon päällä
- Web kameran taustan poisto
- Miten neuro itse tehden ? Löytyy valmista opetettuja neuroverkkoja, luonnollisten kilet
- Gpt3 luonnnisen kielten neuroverkko, pelkkä opetukse sähkönlutus 200'
- Kissa koira taso, valmis verkko, ottaa valmis verkko ja opettaa omia , tai valmiiksi suunniteltu, mitä ei ole opetettu, voidaan kirjoittaa itse okkok arkkkitehtuuri,tensorflowlla, voidaan määrittä itse
- Valmiita koodeja, kerätään data , koodataan neuroverkko, optetetaan testataan ja opetetaan , ootetaan robotilla käyttöön

Regressiomallit

- Pisteet X-y ja vedetään viiva mikä arvo todennäköisesti on
- Dense verkko , jokainen arvo layerill' on kytkentty jokaiseen pis
- 100 x100 , kytkeytyy 100 pisteeseen seuraavaan layeriin
- Neuroverkolla voidaan ratkoa monimutkaisia, moniulottiesia riippuvuuksia
- Rajattomat mahdollisuudet: kunnonvalvonta, prosessin ohjaus , tuotannon ohjaus
- Mallit on kevyitä,
- Sensor mesh, laskenta pilvessä
- Vaatii oikeat ratkaisut ja lähtödatan => helppo
- Sensoridataan voi olla tietokannoissa, vois olla vaikea päästä käsiksi



Reinforcement learning



- Vahvistusoppiminen, voi perustua nv tai ei
- Jäljitellään tapaa jolla ihmiset oppivat, syväoppiminen, annetaan pisteitä hyvistä ja vähennetään huonoista, dopamiini
- Tärkeimmät toimijat OpenAI ja DeepMind
- Reward functions, vaatii tehoa opetuksessa, GOO 2017, dota, starcraft, kuvantunnistus luokittele 5 min opetus, objektin tunnistus 1 päivä, kuukausi DRL hyvällä läppärillä

Koneoppimisen kirjastot

- Ei tarvitse tietää matikkaa taustalla
- Tensorflow, google, avoin, hyvin laaja, mallit ja työkalut, versioiden kanssa tarkkana, näytönohjaimen saaminen toimivaksi, voi olla työlästä, versio pilkulleen oikein, dockerit, contit
- Pytorch , pääosin FB
- Darknet , objektien tunnistus , Yolo
- Keras, abstraktiotaso muiden päälle voi käyttää myös pytorch
- Pandas, datan manipulointiin kirjasto, helpottaa datan tuomista koneoppimisen mallille, voi puhdistaa dataa
- Googlen labissa voi käyttää grafiikkakortteja, nvidia ok

robowars

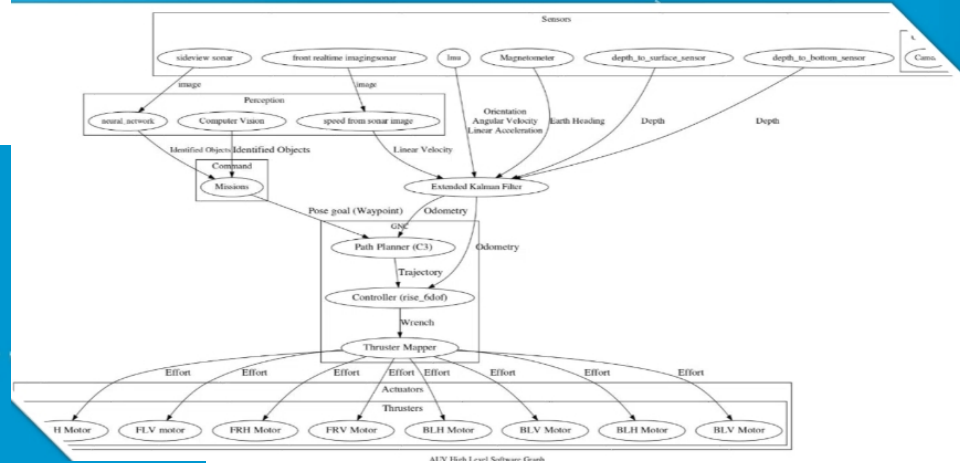
- VM wahren näytönohjaimet
- Linux virtualisointi
- 3x3 areena, yksi kuoppa, autonominen 12 kg (6kg) robotin paikan laskeminen UA datan mukaan
- Vastustajan tunnistus. Kerätään 1000-2000 kuvaa robotista simulaatiossa, labeloidaan kahteen kategoriaan, ei robottia ja robottia ja paikka kuvassa, opetetaan neuroverkko, testataan miten verkko toimii simulaatiossa

Project AUV

(Autonomous underwater

- toimintasyvyys 50m
- 8 potkurilaitetta
- Robot operating system (ROS)
- Kuluttaja luokan kaikuluotaimia
- Monenlaisia Ai algoritmeja.
Perinteisiä/neuroverkkopohjaisia.

HIGH LEVEL SOFTWARE GRAPH



Hyviä linkkejä seurata alan suuntauksia.

LEX FRIDMAN, HAASTATELLUT MONIA ALAN PIONEEREJA

[HTTPS://WWW.YOUTUBE.COM/CHANNEL/UCSHZKYAWB77IXDDSGOG4IWA](https://www.youtube.com/channel/UCSHZKYAWB77IXDDSGOG4IWA)

TWO MINUTE PAPERS, NOPEITA KATSAUKSIA UUSIIN TIETEELLISIIN JULKAISUIHIN, PÄÄASSISSA TEKOÄLYYN LIITTYEN.

[HTTPS://WWW.YOUTUBE.COM/CHANNEL/UCBFYPYITQ-7L4UPOX8NVCTG](https://www.youtube.com/channel/UCBFYPYITQ-7L4UPOX8NVCTG)

ROBERT MILES, TEKOÄLY TURVALLISUUTEN LIITYVIÄ VIDEOITA

[HTTPS://WWW.YOUTUBE.COM/CHANNEL/UCLB7AZTWC6VFZRBSO2UCBMG](https://www.youtube.com/channel/UCLB7AZTWC6VFZRBSO2UCBMG)

2.2

- Opetus, ei liity vielä rossiin
- `pip3 install pandas --user`
-

Toisen luennon tentti

Neuroverkon opetukseen tarvitaan aina GPU

Valitse yksi:

- ☐ Tosi
- ☒ Epätosi

Relu on suosittu aktivointi funktio koska:

Valitse yksi tai useampi:

- ☐ a. Se antaa yleensä paremman tuloksen kuin muut aktivointifunktiot
- ☐ b. Se on esimerkiksi tensorflow kirjastossa ainoa valmiina saatava aktivointifunktio
- ☒ c. Aktivointi funktion arvo ei ole rajoittunut yhteen
- ☒ d. Se on laskennallisesti kevyempi kuin monet muut aktivointifunktiot

Kuvantunnistuksessa pitää aina käyttää konvoluutio neuroverkkoa.

Valitse yksi:

- ☐ Tosi
- ☒ Epätosi

Neuroverkolle syötettävä data koitetaan yleensä saada -1 ja 1 välille, jotta muistin käyttö olisi mahdollisimman pientä

Valitse yksi:

- ☐ Tosi
- ☒ Epätosi

sigmoid funktio on melko usein käytetty optimoija "optimizer"

Valitse yksi:

☐ Tosi

☒ Epätosi

tensorflow:ssa epochs parametri määrittelee kuinka monta kertaa koko datasetti käydään läpi opetuksessa.

Valitse yksi:

☒ Tosi

☐ Epätosi

opetusdaten tallennus