CLML

Cosmology with Large scale structure using Machine Learning

2022.03.04

outline

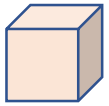
❖ I'm estimating cosmological parameters(Ω_m, σ_8) using 3dCNN

arXiv:1908.10590

From last meeting..

❖ I used only xyz position of halos from PINOCCHIO(simulation code)
❖ I added information about mass and velocity for the input data

About data



(32,32,32)

Halo's xyz position → Numpy.histogramdd → **the number of halos** in each bin

Halo's xyz position
weight = halo's mass → Numpy.histogramdd → **the sum of mass** in each bin

Halo's xyz position
weight = halo's speed
(Speed = $\sqrt{{v_x}^2 + {v_y}^2 + {v_z}^2}$) → Numpy.histogramdd → **the sum of speed** in each bin

Normalization

the number of halos
in each bin

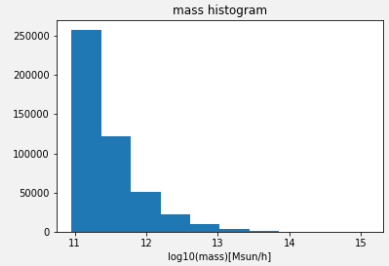
bins_nhalo

norm_pos = bins_pos/max(bins_pos)

the sum of mass
in each bin

bins_mass

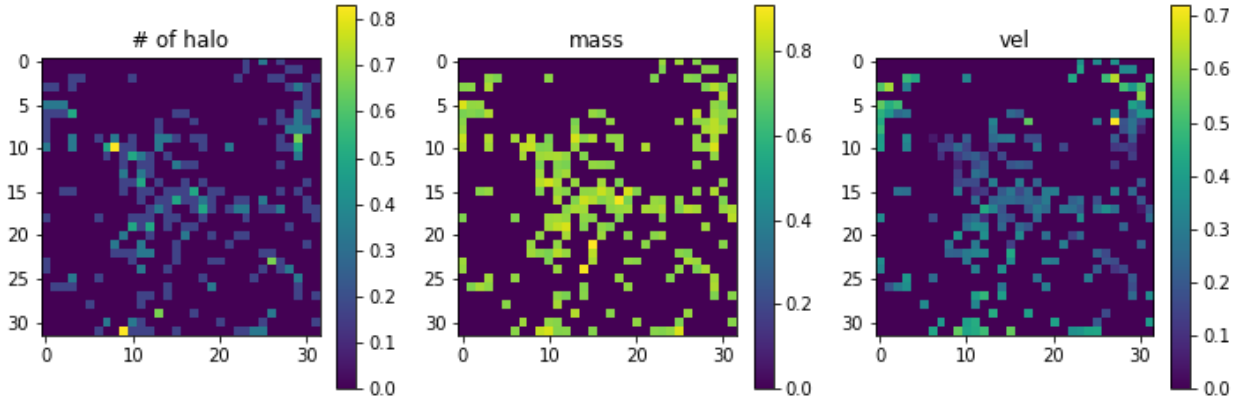
bins_mass = np.log10(bins_mass)
bins_mass[bins_mass == -np.inf] = 0 (change -∞ to 0)
norm_bins_mass = bins_mass/15



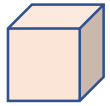
the sum of speed
in each bin

bins_speed

ave_bins_vel = bins_vel/bins_pos
ave_bins_vel = np.nan_to_num(ave_bins_vel) (change num to 0)
norm_bins_vel = ave_bins_vel/max(ave_bins_vel)



DATA TYPE



Shape = (32,32,32,1)

bins_Nhalo

bins_mass

bins_speed

Shape = (32,32,32,2)

bins_Nhalo

bins_mass

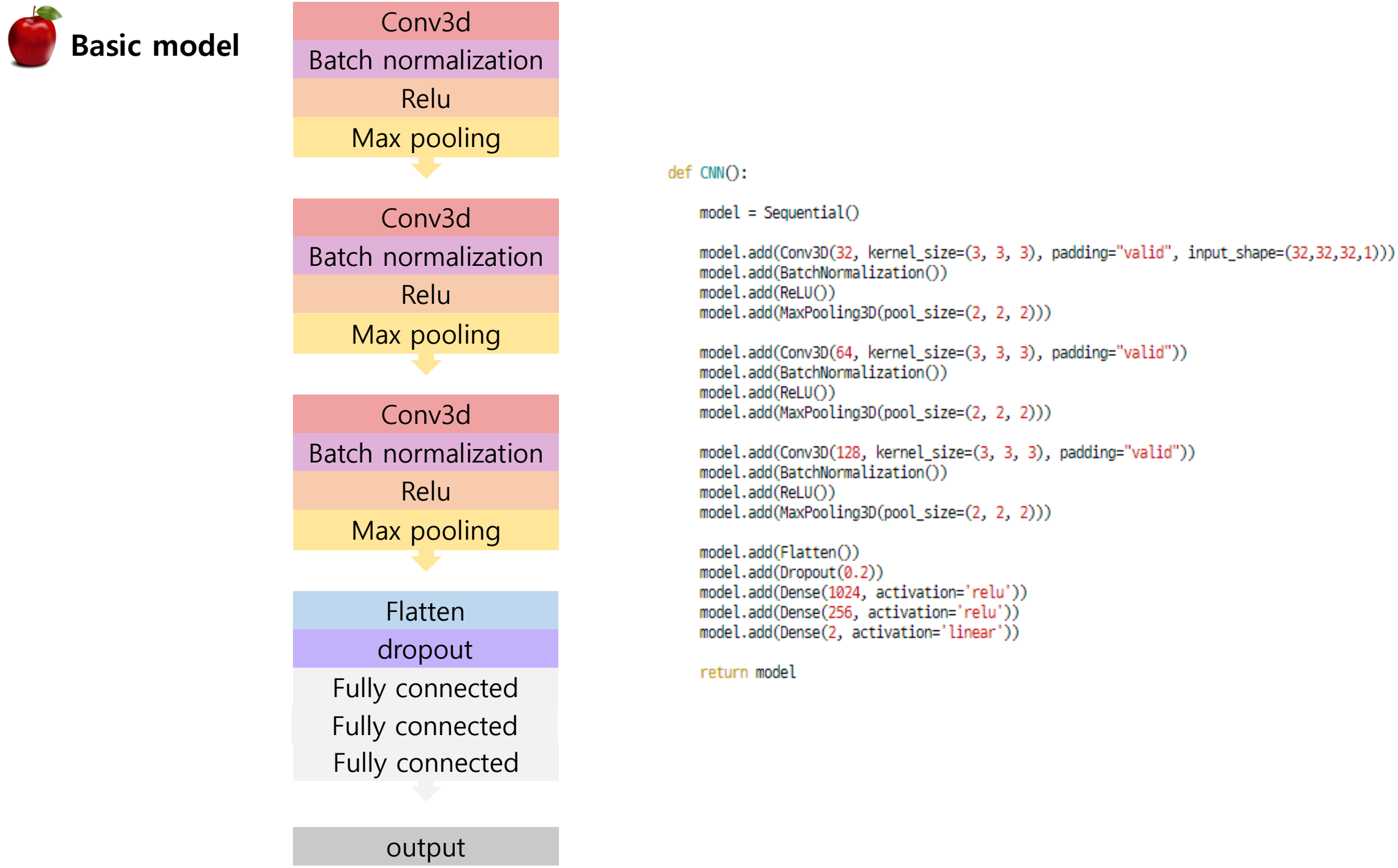
Shape = (32,32,32,3)

bins_Nhalo

bins_mass

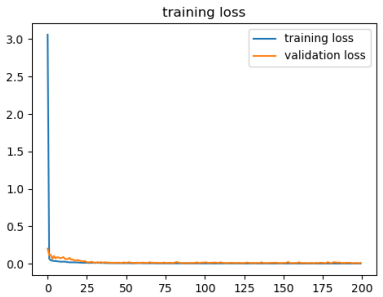
bins_speed

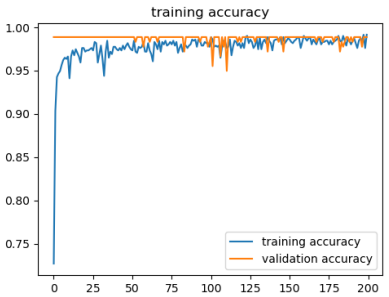
Basic model

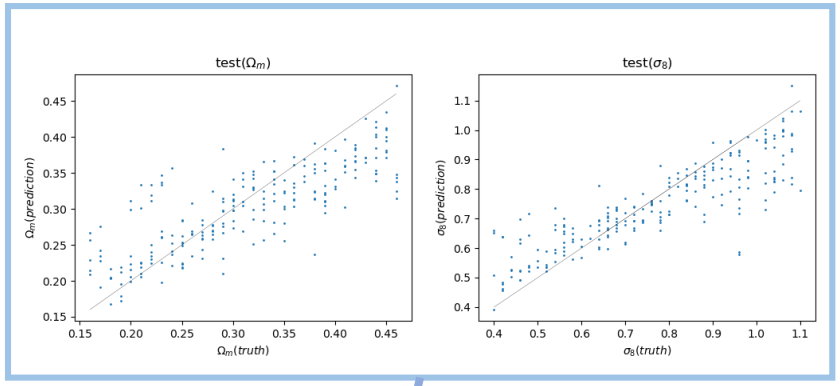


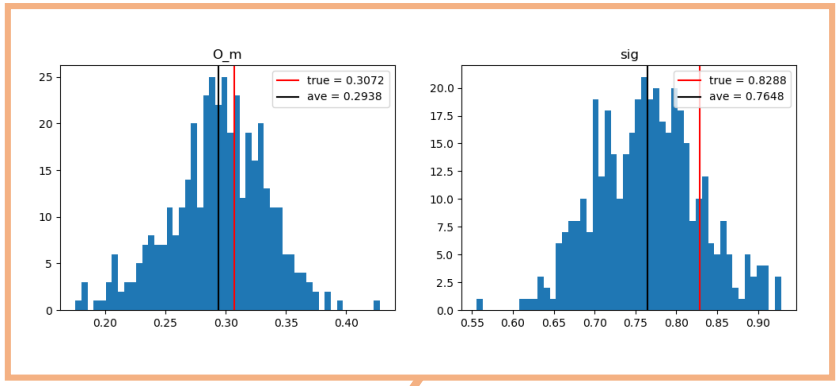

```python
def CNN():

    model = Sequential()

    model.add(Conv3D(32, kernel_size=(3, 3, 3), padding="valid", input_shape=(32,32,32,1)))
    model.add(BatchNormalization())
    model.add(ReLU())
    model.add(MaxPooling3D(pool_size=(2, 2, 2)))

    model.add(Conv3D(64, kernel_size=(3, 3, 3), padding="valid"))
    model.add(BatchNormalization())
    model.add(ReLU())
    model.add(MaxPooling3D(pool_size=(2, 2, 2)))

    model.add(Conv3D(128, kernel_size=(3, 3, 3), padding="valid"))
    model.add(BatchNormalization())
    model.add(ReLU())
    model.add(MaxPooling3D(pool_size=(2, 2, 2)))

    model.add(Flatten())
    model.add(Dropout(0.2))
    model.add(Dense(1024, activation='relu'))
    model.add(Dense(256, activation='relu'))
    model.add(Dense(2, activation='linear'))

    return model
```

Figure explanation





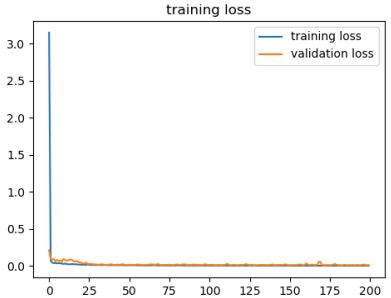


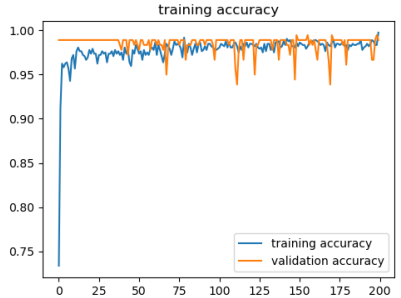


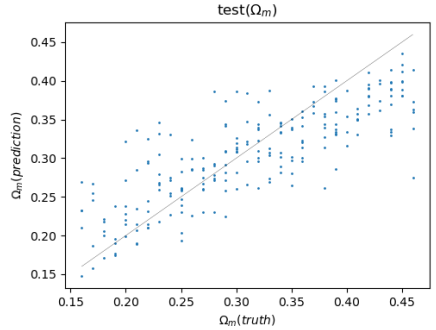
X axis is true value and y axis is prediction value

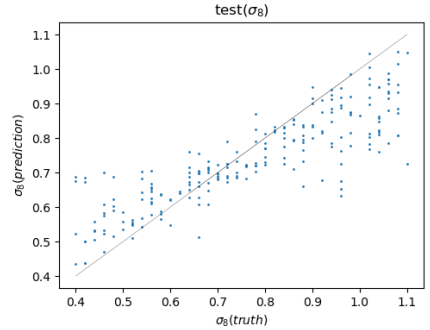
For this plot,
DATA I used 500 single cosmology(Ω_m= 0.3072, σ_8=0.8288) with different random seed
MODEL I used tensorflow's ModelCheckpoint which is lowest loss during the training

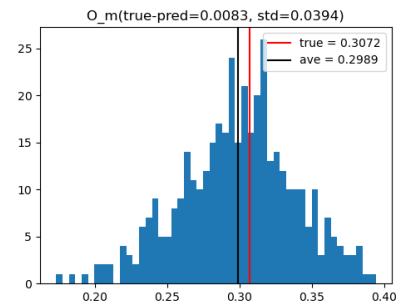
bins_Nhalo

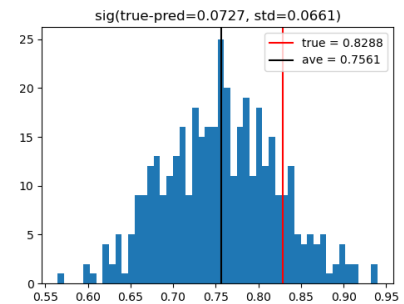




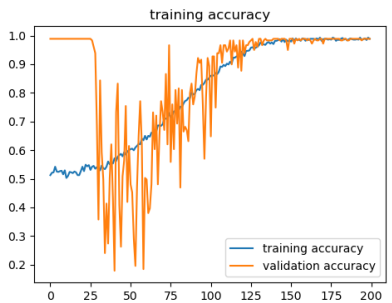


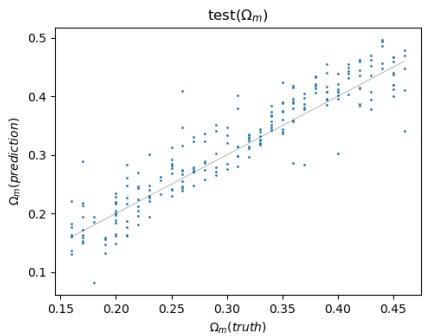


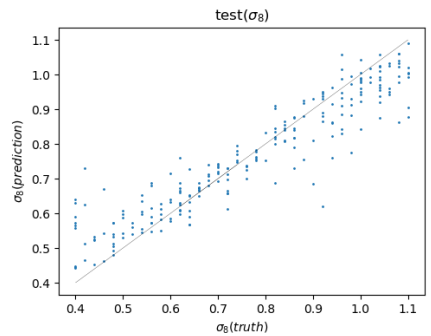


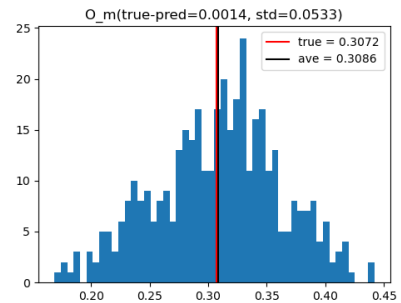


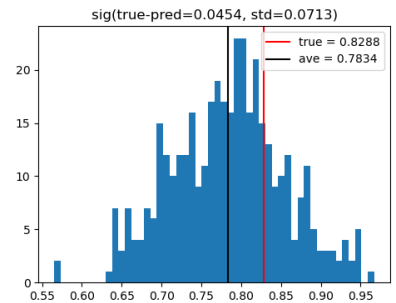
bins_mass



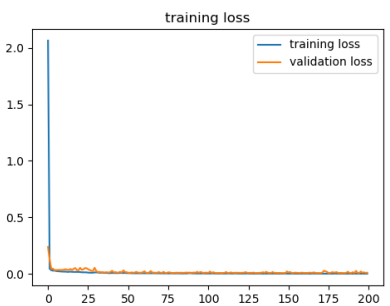


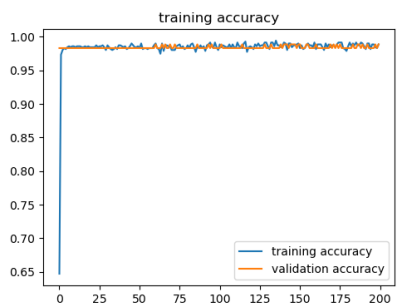


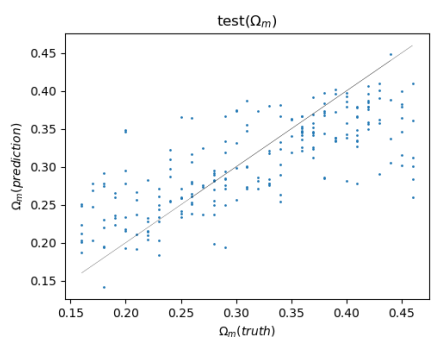


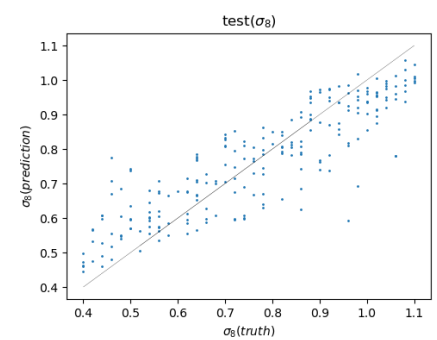


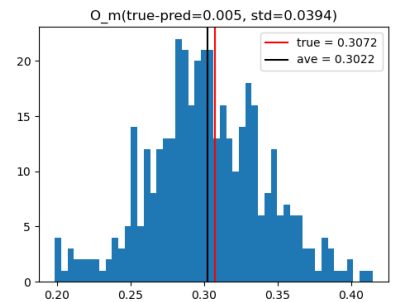
bins_speed

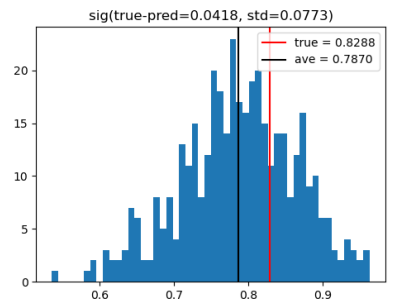






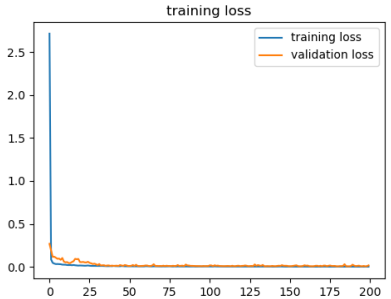


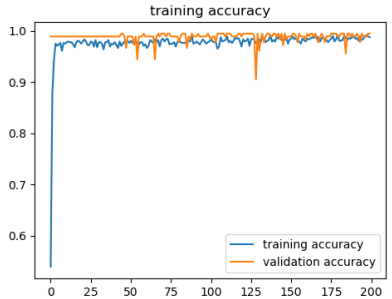


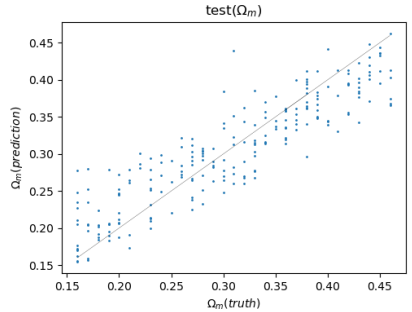


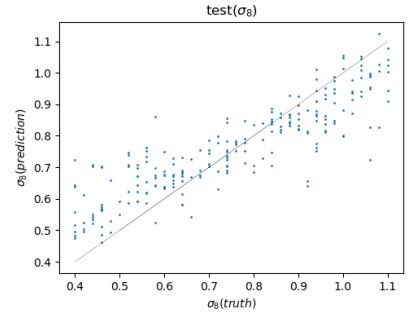
bins_Nhalo

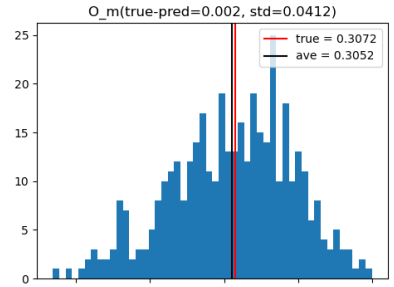
bins_mass

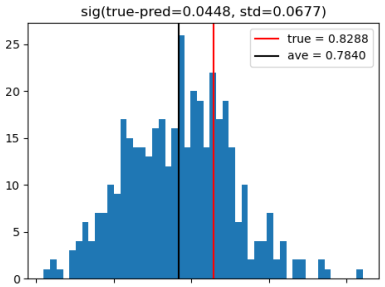








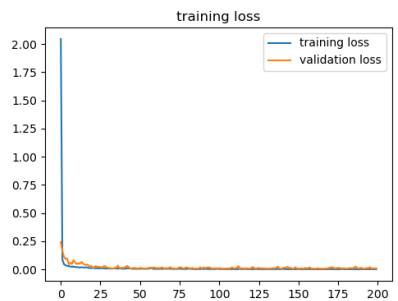


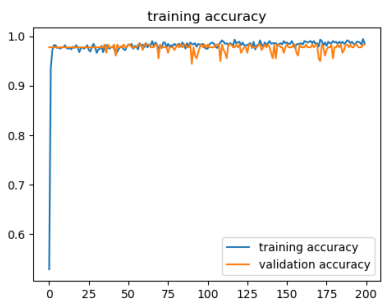


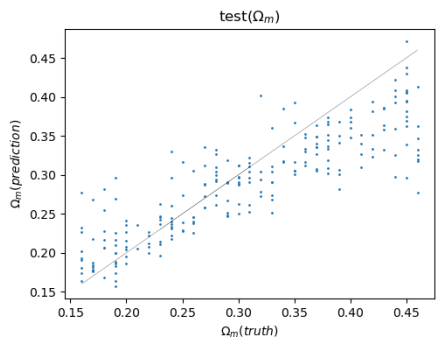
bins_Nhalo

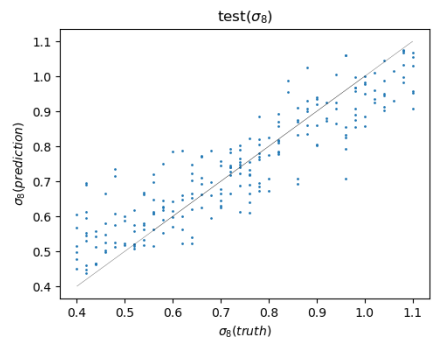
bins_mass

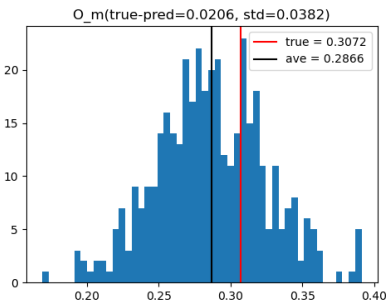
bins_speed

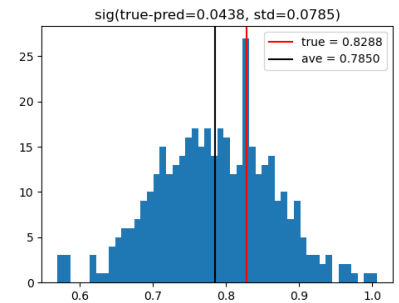




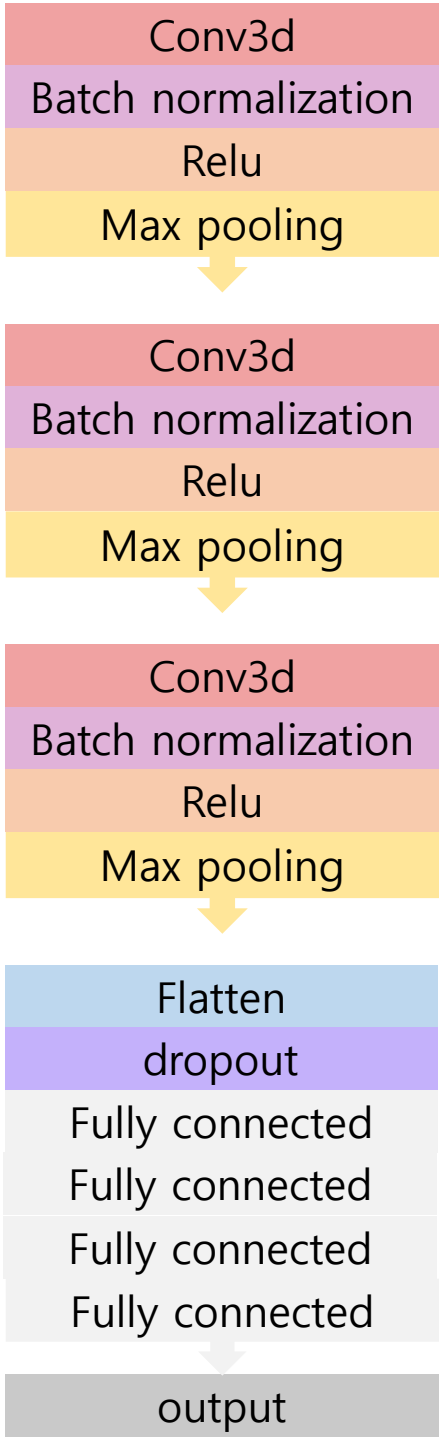






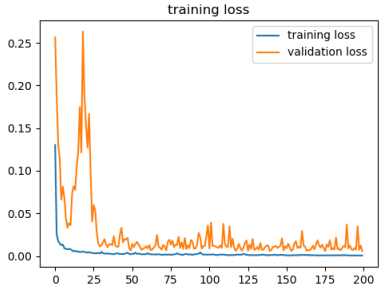


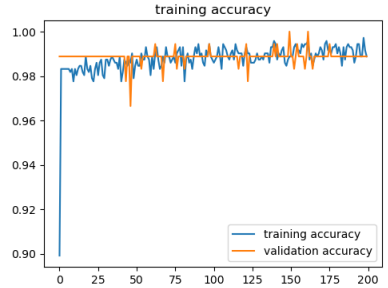
model

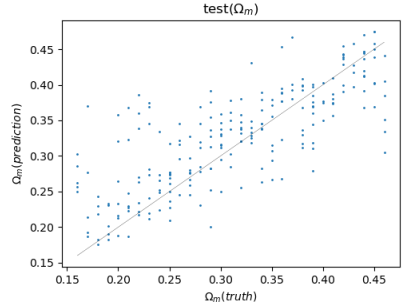


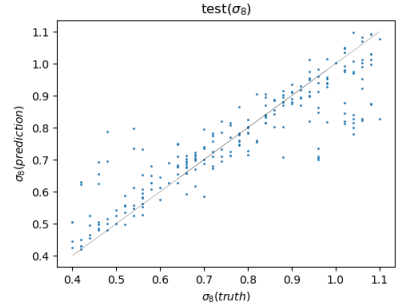

```python
def CNN():

    model = Sequential()

    model.add(Conv3D(32, kernel_size=(3, 3, 3), padding="valid", input_shape=(32,32,32,3)))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(AveragePooling3D(pool_size=(2, 2, 2)))

    model.add(Conv3D(64, kernel_size=(3, 3, 3), padding="valid"))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(AveragePooling3D(pool_size=(2, 2, 2)))

    model.add(Conv3D(128, kernel_size=(3, 3, 3), padding="valid"))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(AveragePooling3D(pool_size=(2, 2, 2)))

    model.add(Flatten())
    model.add(Dropout(0.2))
    model.add(Dense(1024))
    model.add(Activation('relu'))

    model.add(Dense(256))
    model.add(Activation('relu'))

    model.add(Dense(512))
    model.add(Activation('relu'))

    model.add(Dense(2))
    model.add(Activation('linear'))

    return model
```

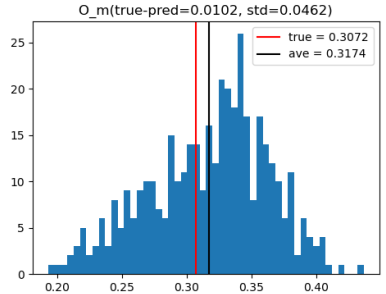
bins_Nhalo

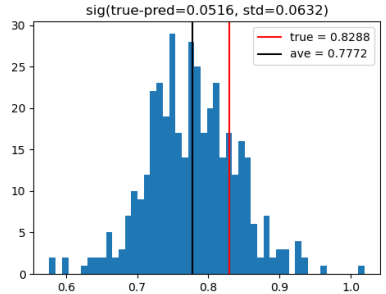




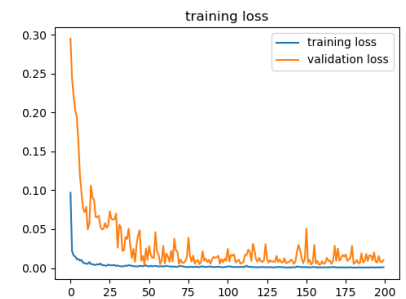


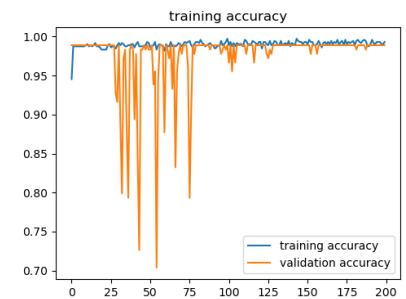


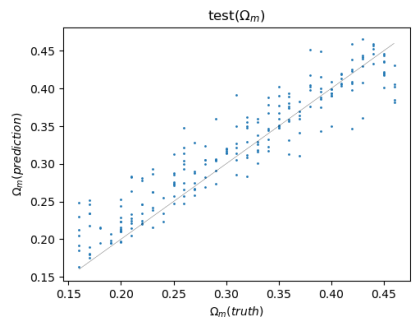


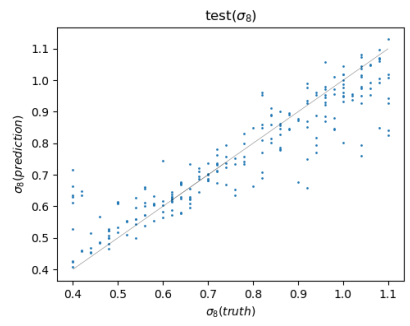


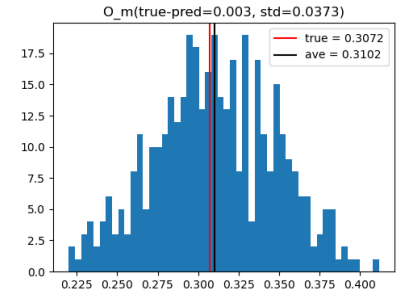
bins_mass

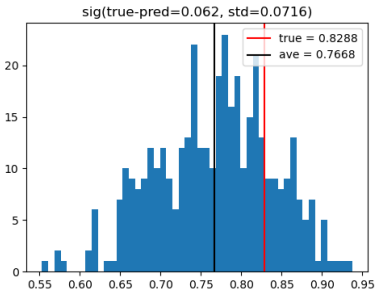




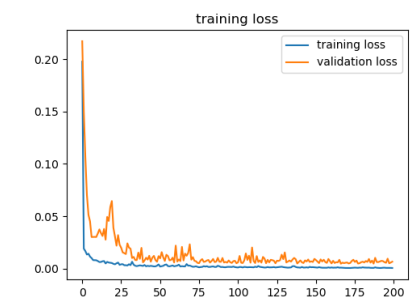


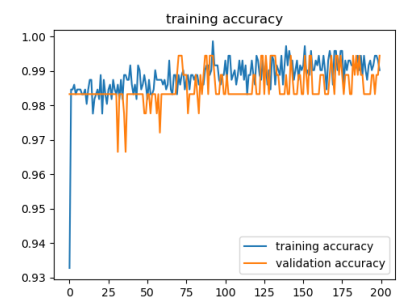


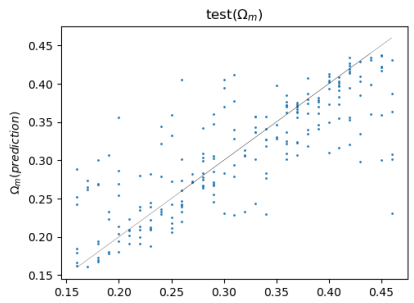


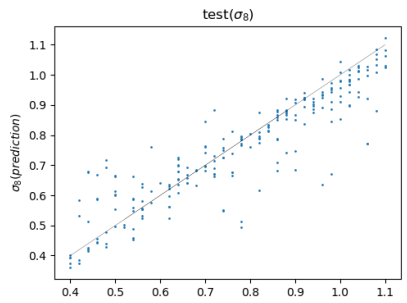


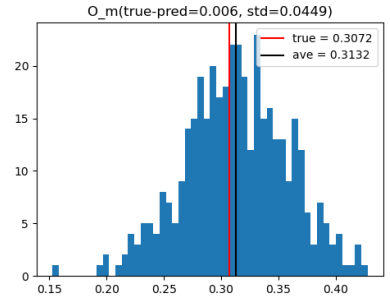
bins_speed

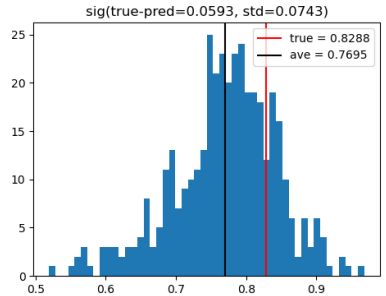






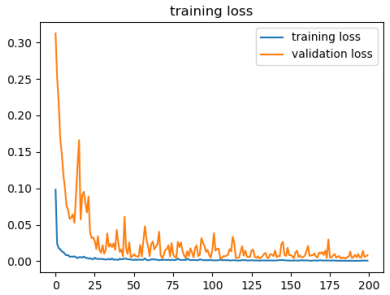


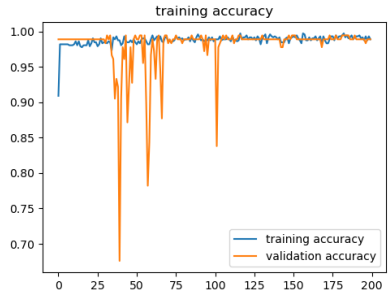


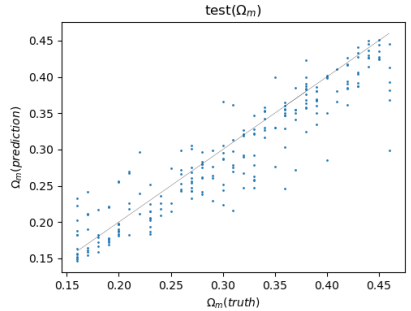


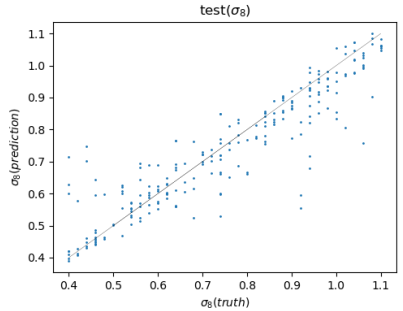
bins_Nhalo

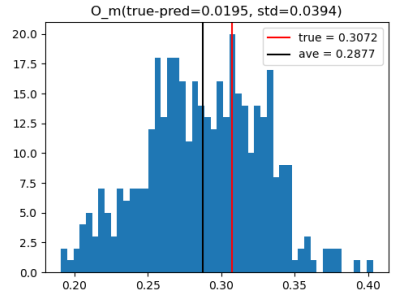
bins_mass

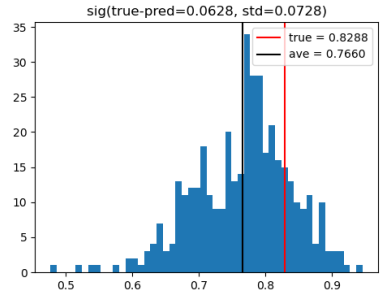








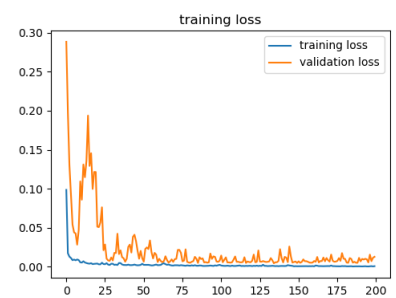


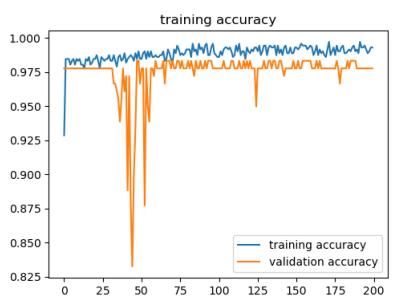


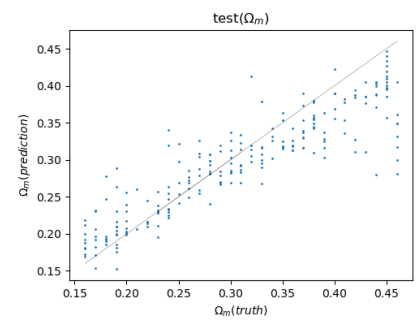
bins_Nhalo

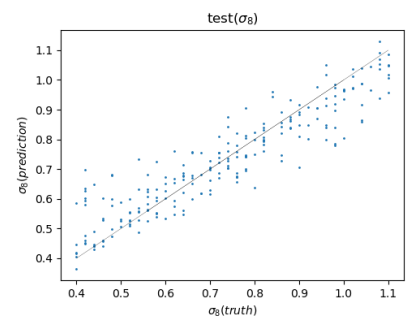
bins_mass

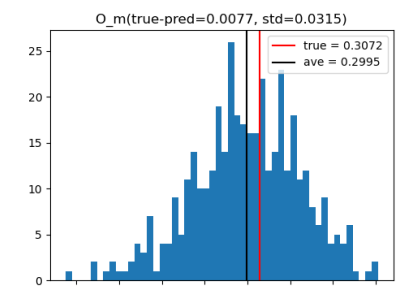
bins_speed

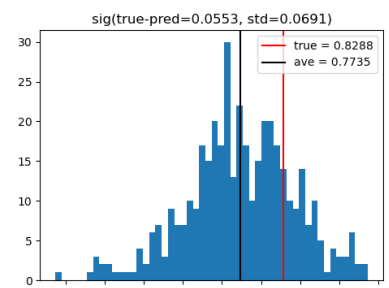












result

	apple				banana			
	Om		sig		Om		Sig	
	True-pred	std	True-pred	std	True-pred	std	True-pred	std
nhalo	0.0083	0.0394	0.0727	0.0661	0.0102	0.0462	0.0516	0.0632
mass	0.0014	0.0533	0.0454	0.0713	0.003	0.0373	0.062	0.0716
vel	0.005	0.0394	0.0418	0.0773	0.006	0.0449	0.0593	0.0743
nhalo+mass	0.002	0.0412	0.0448	0.0677	0.0195	0.0394	0.0628	0.0728
nhalo+mass+vel	0.206	0.0382	0.0438	0.0785	0.0077	0.0315	0.0553	0.0691

Future..

- Add more cosmological parameters to predict
- Change the architecture of model