ML in Fundamental Physics

Ludwig-Maximilians-Universität München
Prof. Dr. Dieter Lüst, Dr. Sven Krippendorf, Summer Term 2021

Exam

- You can start working as soon as you get the exam via email.
- You are not allowed to communicate with other people during the exam. You are allowed to consult your notes and online coding documentations to assist your answers.
- If you want to label your answers, please use the exam ID and NOT your name on each sheet.
- The deadline for the submission of your exam solutions is 19.00 via Moodle. This time includes the time it takes to generate a **single** .pdf of your solutions and to upload it.
- Your code answers need to be submitted in a single jupyter notebook .ipynb.
- You need to fill in the confirmation and authentification cover sheet which can be found in the exam section on the Moodle website. This has to be submitted as part of your exam.
- During the exam you need to be present in our standard Zoom meeting for the lectures (Meeting ID: 976 4541 2127,Passcode: 228755). Your camera needs to be turned on and your microphone needs to be turned on. We will open the meeting from 15:30 onwards such that we can perform technical checks.
- In case you encounter connection issues and you drop out of the call. Please call 089-2180-4368 to report your issues.
- In case you have questions, please use the raise hand and chat function in the first instance.

1 Loss components *(5 points)*

- *(2 points)* You are fitting data with a machine learning regression model. Which three (structural) components does the expected loss have (names are sufficient)?

  _____

  _____

  _____

- *(2 points)* Sketch a typical example of training and validation loss showing overfitting. (a labeled diagram suffices)

- *(1 point)* In a classification problem, how can I assess that my network's performance is above that of pure guessing by looking at the test accuracy?

  _____

2 Comparison of two optimisers *(10 points)*

- How are parameters updated using stochastic gradient descent and stochastic gradient decent with momentum? (2 points)
- Using the example $f(x) = x^2$, show that for a learning rate of $\eta > 1$ gradient descent diverges. You can use a numerical example. (4 points)
- Show that using momentum, there exists a range of learning rates $\eta > 1$ where gradient descent using momentum does not diverge. It is sufficient to show this with at least two examples in this range. (4 points)

3 Neural Network *(10 points)*

The following example shows the example of a simple neural network.

In [ ]:
```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt

from tensorflow.keras.models import  Model
from tensorflow.keras.layers import Dense, Input

from tensorflow.python.keras.losses import categorical_crossentropy, binary_crossentropy
from tensorflow.python.keras.optimizers import Adam


num_classes=2

input_network = Input(shape=(10,))
x = Dense(20, activation='tanh')(input_network)
y = Dense(num_classes, activation='sigmoid')(x)

model= Model(input_network,outputs=y)
model.compile(loss=binary_crossentropy,optimizer=Adam(), metrics=['accuracy'])

batch_size=50
epochs=20
model.fit(x_train, y_train,batch_size=batch_size,epochs=epochs,verbose=1,validation_data=(x_test, y_test))
```

- *(4 points)* Write down explicitly the function which the neural network calculates. (Hint: You have to define every activation function you use.)
- *(3 points)* How is the loss calculated during training? Your answer should include an explicit formula with the adapted numerical values.
- *(3 points)* How can the hidden dense layer be modified such that the weights are initialised with a Gaussian with mean zero and standard deviation 100? Provide your answer in python code which in principle can be executed.

4 Short questions *(20 points)*

- *(6 points)* Name an option on how to improve the classification in the following cases. Give a reason why you expect your modification to improve the classification:

  1. XOR with a neural network without a hidden layer.
  2. SVM without a kernel.
  3. Decision tree with a given depth.

- *(3 points)* Name three different types of dimensional data reduction.

- *(2 points)* Which of the following methods is extremising the variational free energy:

  1. Autoencoder
  2. Boltzmann machine
  3. Hopfield network
  4. Generative adversarial network

- *(2 points)* How can I bias my neural network architecture in comparison to a dense feed-forward neural network when trying to classify the types of galaxies shown on images? Name two distinct ways except regularisation (e.g. via L1/L2-losses).

- *(3 points)* Explain how a convolutional layer can be embedded in a standard dense layer. What are the differences to a dense layer?

- *(4 points)* Which type of biased networks are promising for the following type of data structures:

  1. time series data
  2. data with rotational symmetry
  3. data with translation symmetry
  4. data with permutation symmetry

5 Neural Network *(10 points)*

- *(9 points)* Write down an autoencoder using Keras which can be used for 2D Ising spin configurations (40x40). Your network has to have 100000 trainable parameters. Submit your network as an executable code. (Hint: Layers without a bias might render matching the exact number of parameters easier.)
- *(1 point)* How can I augment my training data for improving the network without having to run another MCMC to generate samples?

6 Unsupervised learning *(12 points)*

- *(4 points)* Describe the standard loss and training procedure for a GAN.
- *(6 points)* Provide an executable python implementation of the training step where you make two gradient updates on the training data for the generator when the generator is only updated once.
- *(1 point)* Why is the balance between the generator and discriminator training important for successful training?
- *(1 point)* What is a physics example where a GAN can be useful?

7 ODE and custom loss *(12 points)*

Provide answers to the following questions as executable python code

- *(6 points)* Implement a custom loss function which ensures that the following differential equation is satisfied

$$x' = \sqrt{1+x^2}\, x$$

- *(3 points)* Write a training step which uses this custom loss function evaluated with 1000 uniformly sampled points in the interval (-1,1) for training.
- *(3 points)* Write a training loop where this training step is applied 100 times and output the training error for each training step.