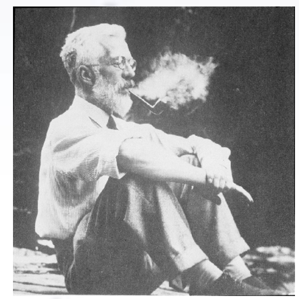
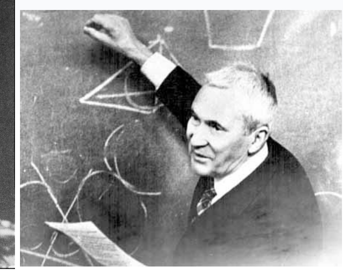
Applied ML

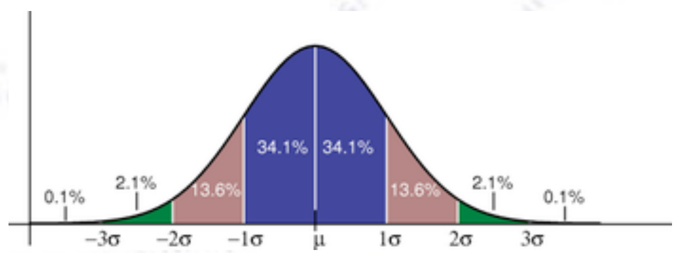
Loss values





Troels C. Petersen (NBI)

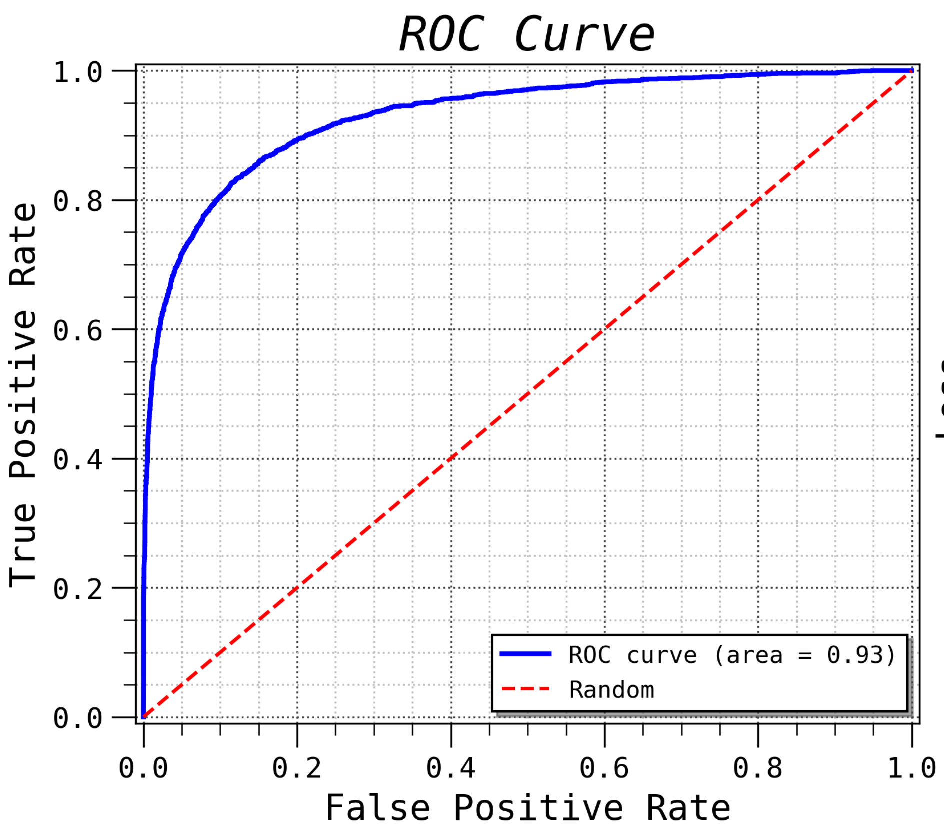


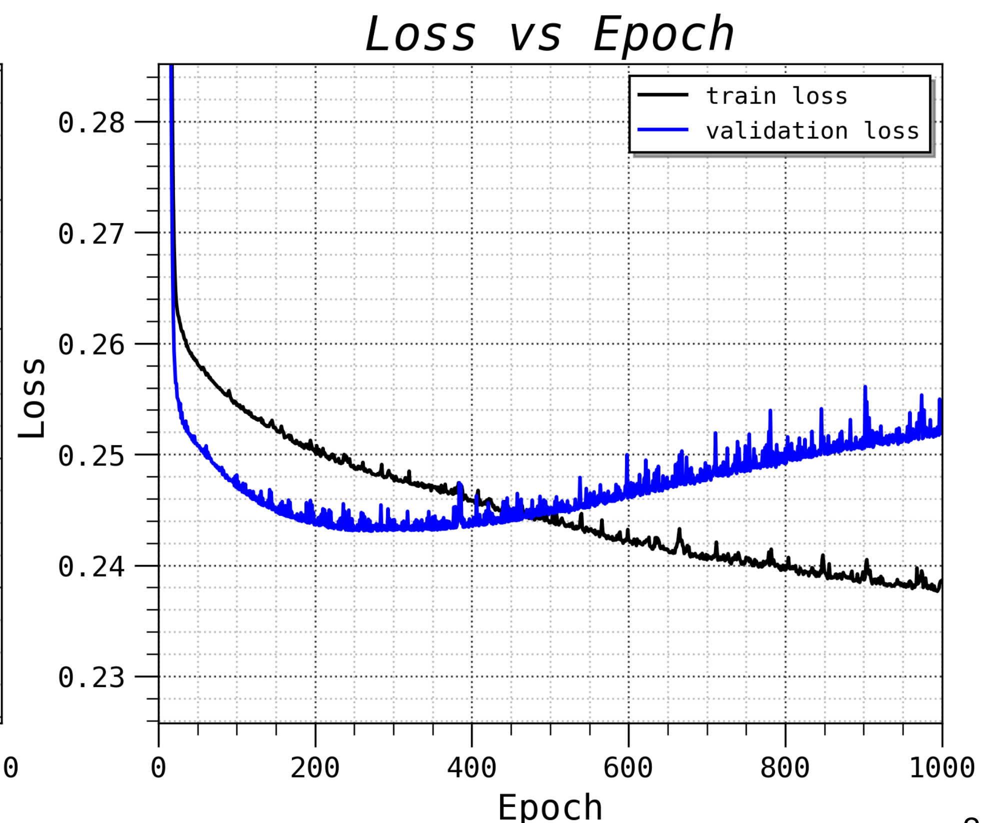
"Statistics is merely a quantisation of common sense - Machine Learning is a sharpening of it!"

Values of the Loss Function

During training (or afterwards!) one can monitor the value of the loss function (L) for the training and validation sample, see below right (here for an NN).

You should understand the shape of the two below figures and their relations.





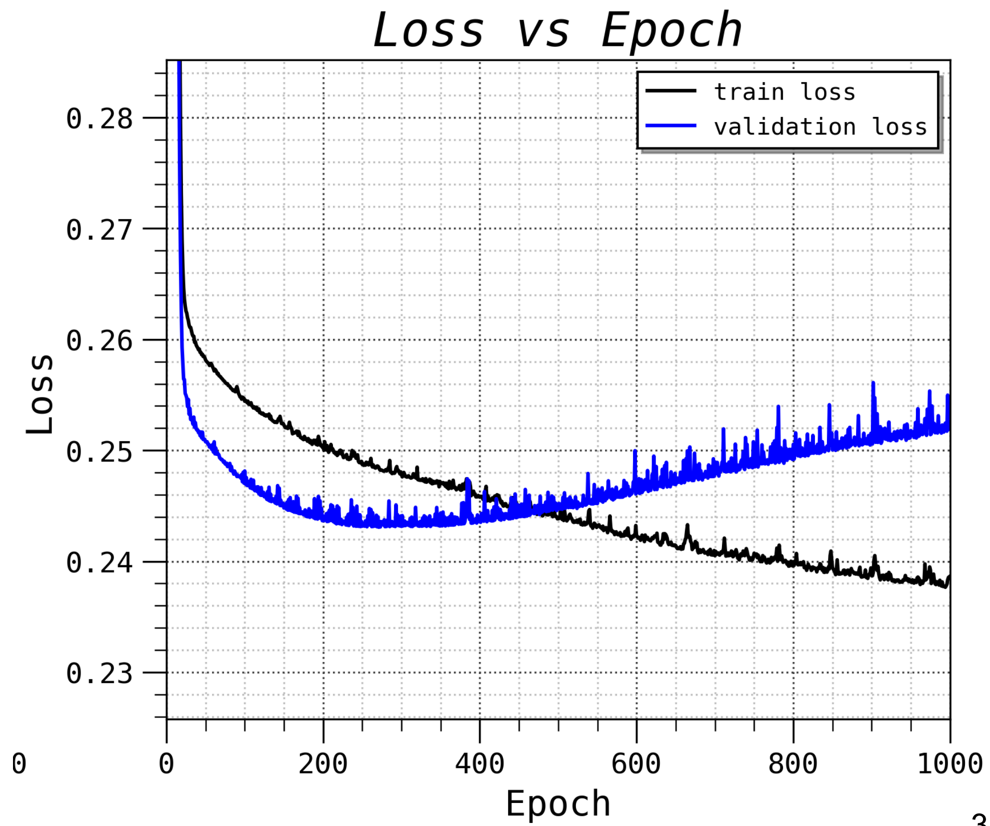
Values of the Loss Function

During training (or afterwards!) one can monitor the value of the loss function (L) for the training and validation sample, see below right (here for an NN).

You should understand the shape of the two below figures and their relations.

<u>Ask yourself the following questions:</u>

- What is the **optimal** Epoch number to stop training at?
- Why does the validation loss start to **increase** after this point?
- Will the training loss **reach zero** for an NN? For a BDT?
- Why is the validation loss **lower** than the training loss?
  (There can be several reasons for this, see the following slides).



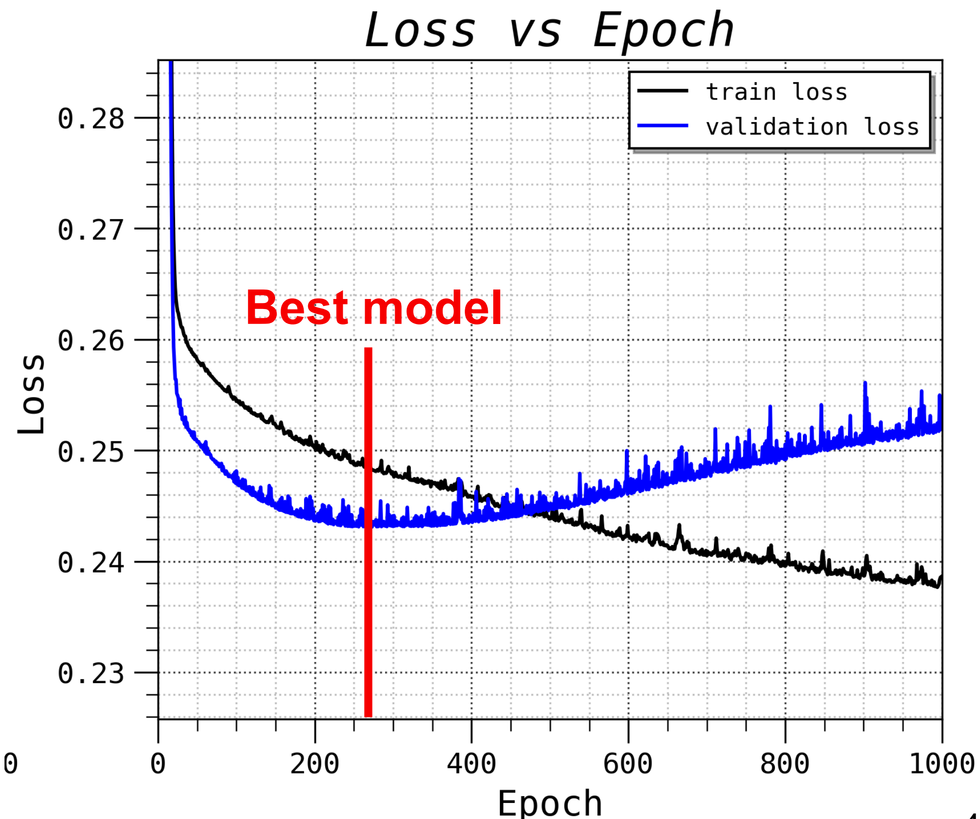
Values of the Loss Function

During training (or afterwards!) one can monitor the value of the loss function (L) for the training and validation sample, see below right (here for an NN).

You should understand the shape of the two below figures and their relations.

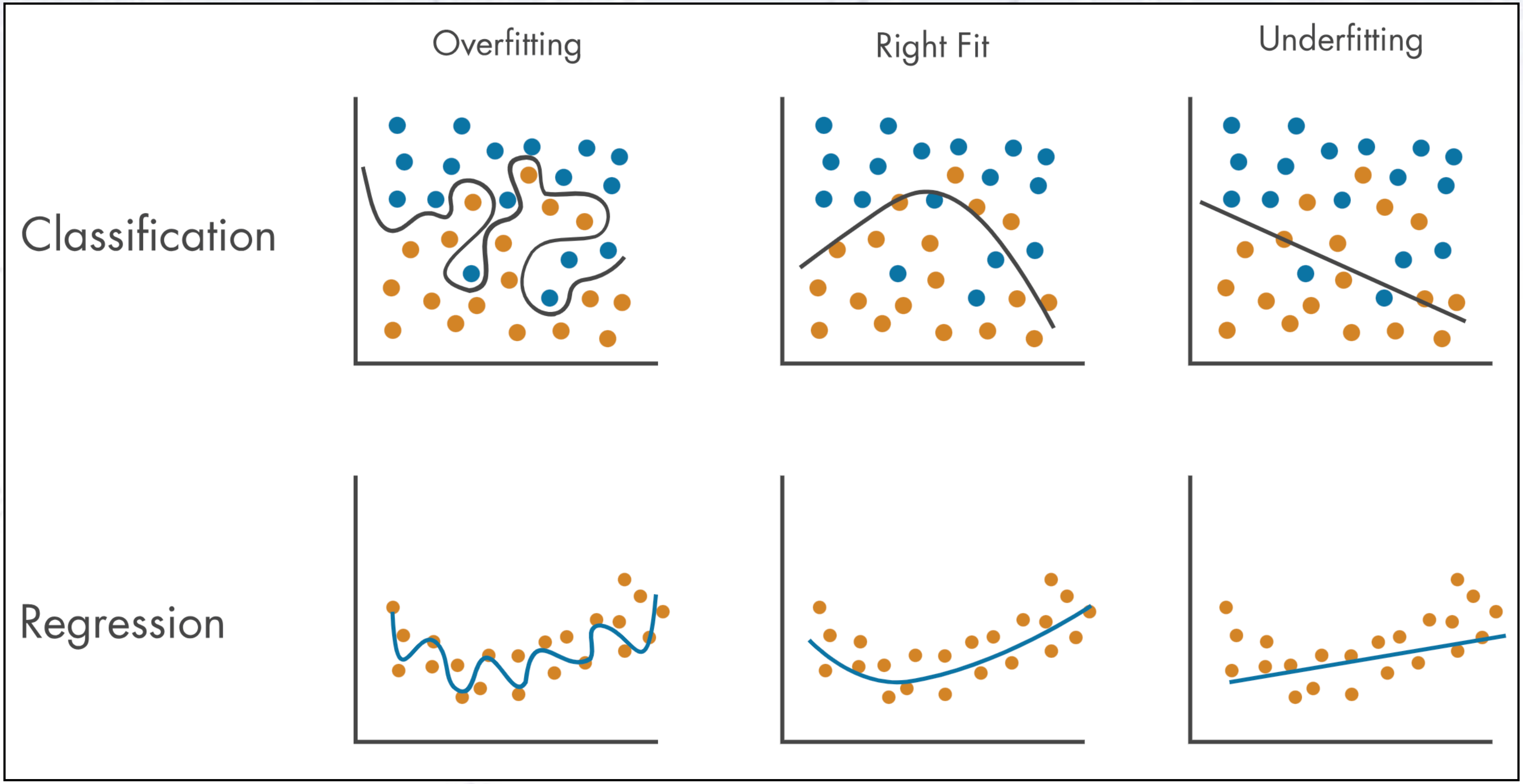
<u>Ask yourself the following questions:</u>

- What is the **optimal** Epoch number to stop training at?
- Why does the validation loss start to **increase** after this point?
- Will the training loss **reach zero** for an NN? For a BDT?
- Why is the validation loss **lower** than the training loss?
  (There can be several reasons for this, see the following slides).



Values of the Loss Function

The validation (and test) loss starts increasing, as overtraining does not yield a result that generalises to other samples.



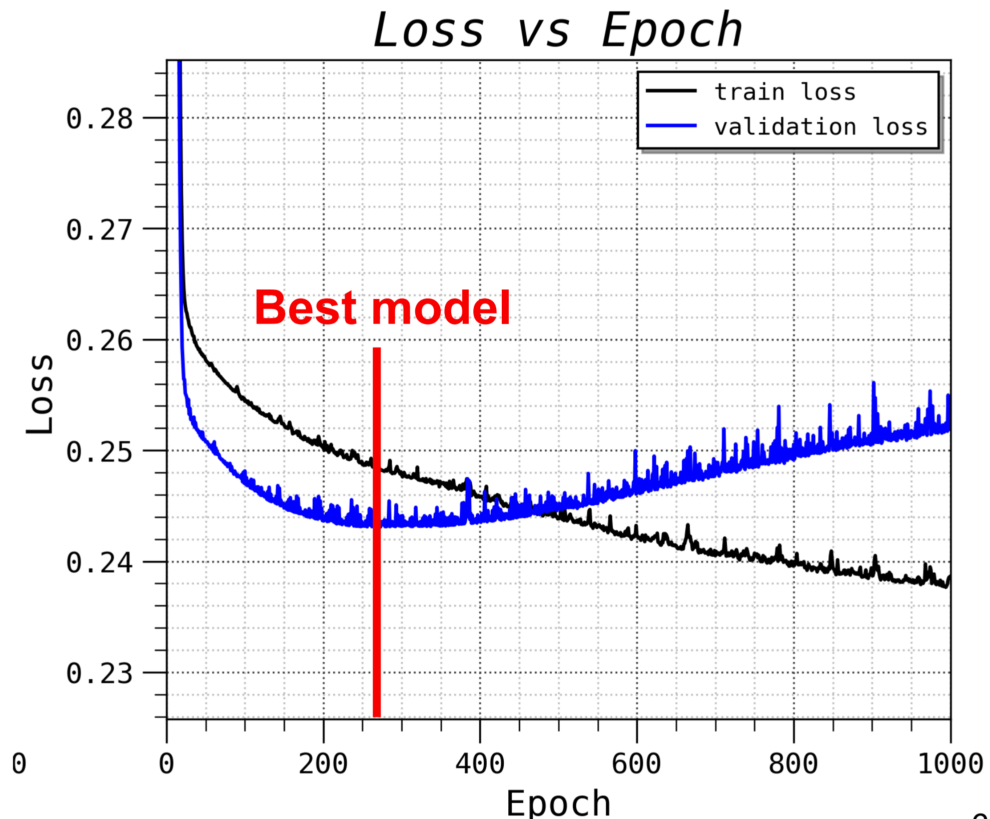
Values of the Loss Function

During training (or afterwards!) one can monitor the value of the loss function (L) for the training and validation sample, see below right (here for an NN).

You should understand the shape of the two below figures and their relations.

<u>Ask yourself the following questions:</u>

- What is the **optimal** Epoch number to stop training at?
- Why does the validation loss start to **increase** after this point?
- Will the training loss **reach zero** for an NN? For a BDT?
- Why is the validation loss **lower** than the training loss?
  (There can be several reasons for this, see the following slides).



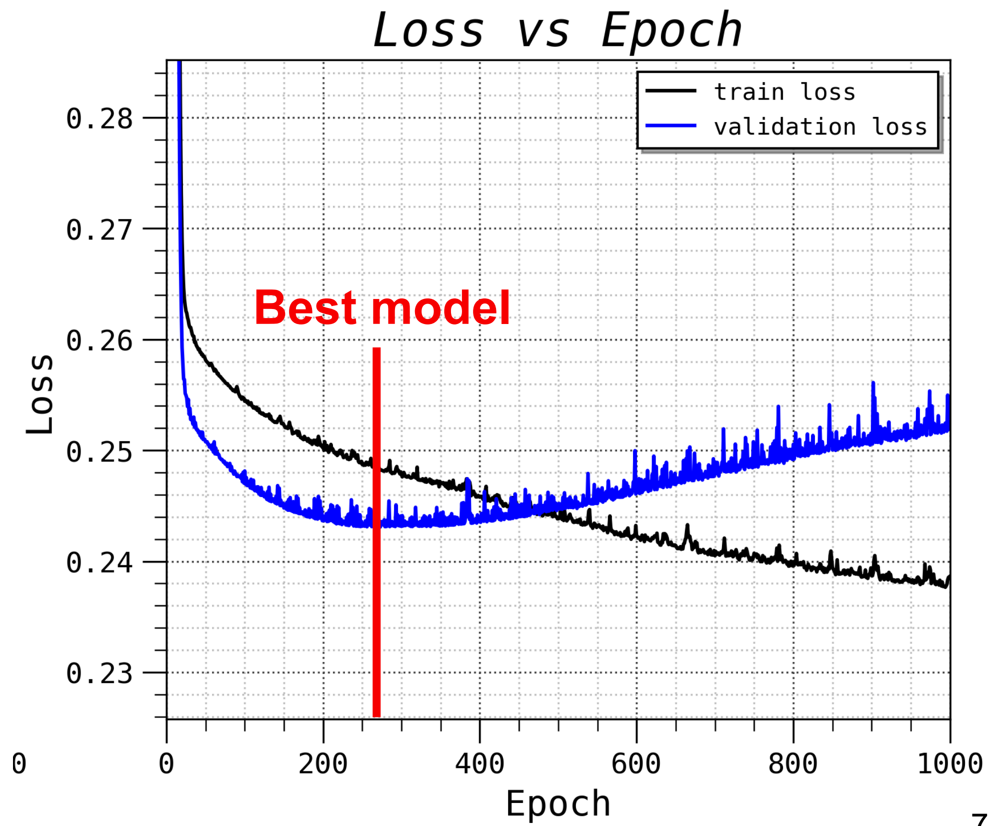
Values of the Loss Function

During training (or afterwards!) one can monitor the value of the loss function (L) for the training and validation sample, see below right (here for an NN).

You should understand the shape of the two below figures and their relations.

<u>Ask yourself the following questions:</u>

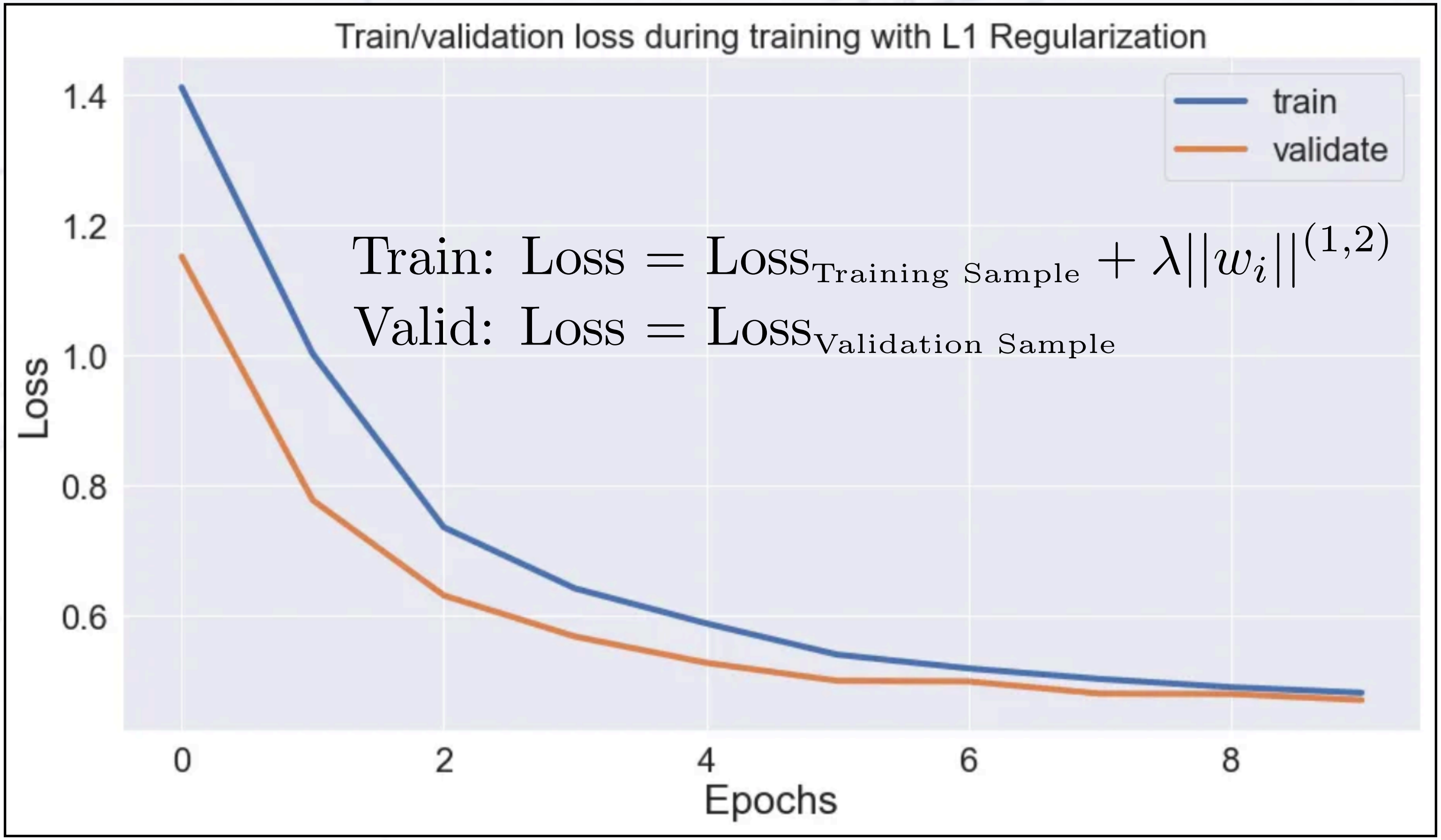
- What is the **optimal** Epoch number to stop training at?
- Why does the validation loss start to **increase** after this point?
- Will the training loss **reach zero** for an NN? For a BDT?
- Why is the validation loss **lower** than the training loss?
  (There can be several reasons for this, see the following).



Values of the Loss Function

1. Regularization:

During training, the Loss Function calculated on the training sample includes a regularization term (linear or squared), which ensures that the model is well behaved. This term is not applied to the validation sample.

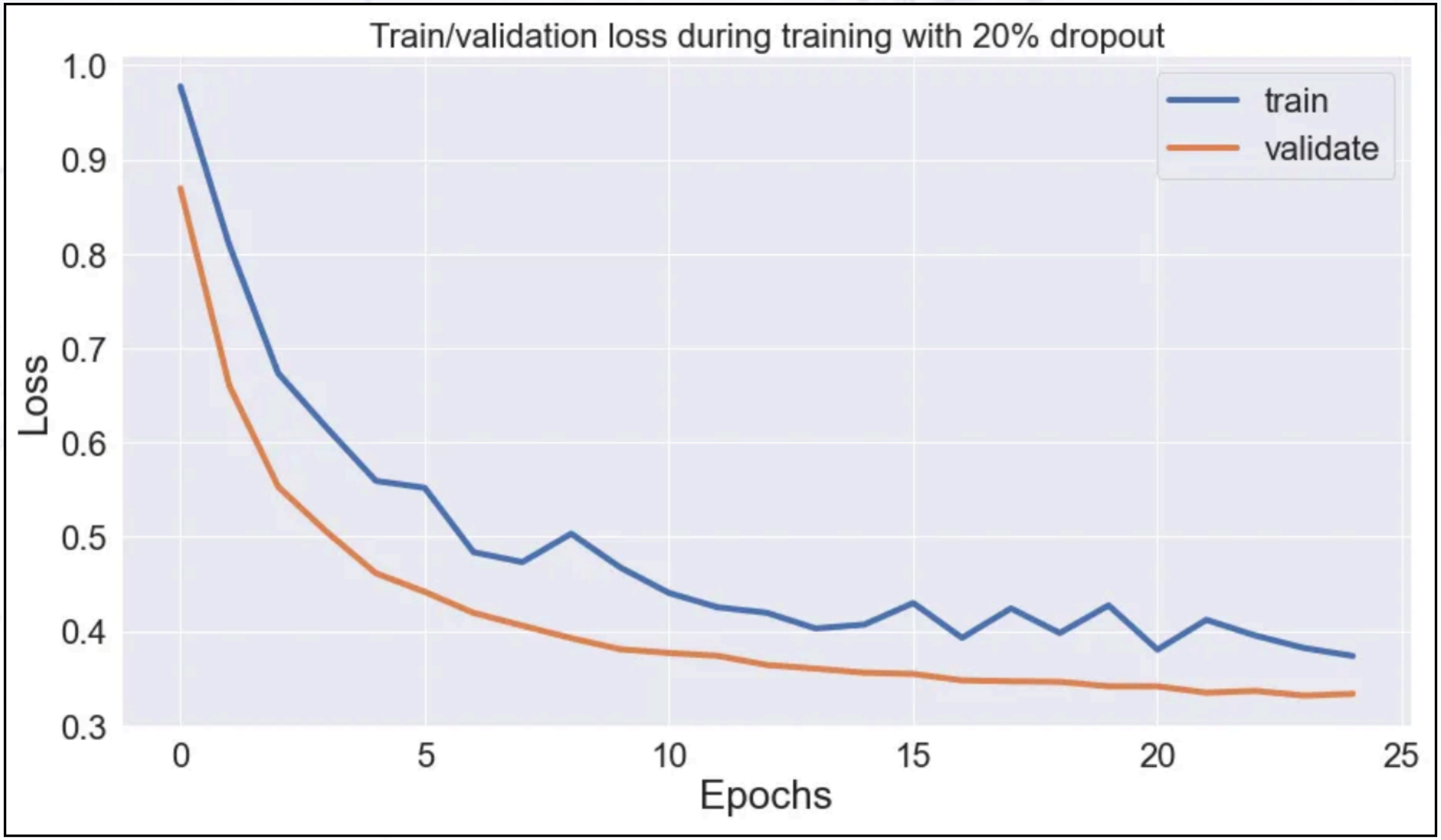


$$\text{Train: Loss} = \text{Loss}_{\text{Training Sample}} + \lambda ||w_i||^{(1,2)}$$

$$\text{Valid: Loss} = \text{Loss}_{\text{Validation Sample}}$$

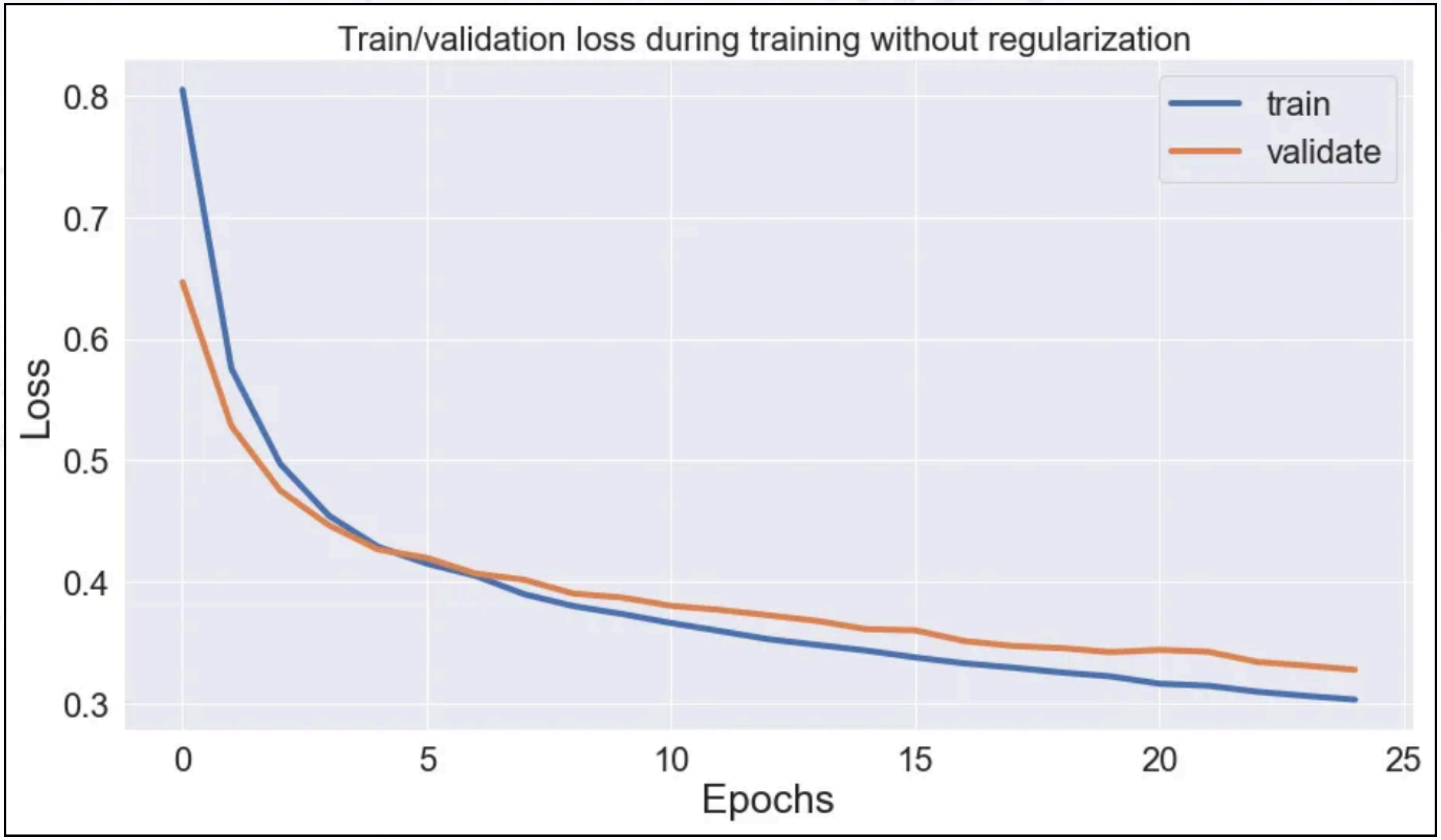
Values of the Loss Function

2. Dropout:
During training of a NN, the algorithms randomly freezing neurons in a layer. This penalises the performance during training, and also gives “spikes” in the training loss.



Values of the Loss Function

3. Evaluation time:
The training loss is calculated as an average over an epoch (for an NN), while the validation loss is calculated at the end. If the model improves significantly during an epoch, the validation loss becomes lower… but only for a while!



Values of the Loss Function

4. Luck!

Occasionally, one simply chooses a "lucky" validation set. This is more likely to happen for small samples than for larger ones. However, this effect can be hard to disentangle from other effects, unless one is in control of these.