Neural Networks 2

Anders Krogh

Center for Health Data Science

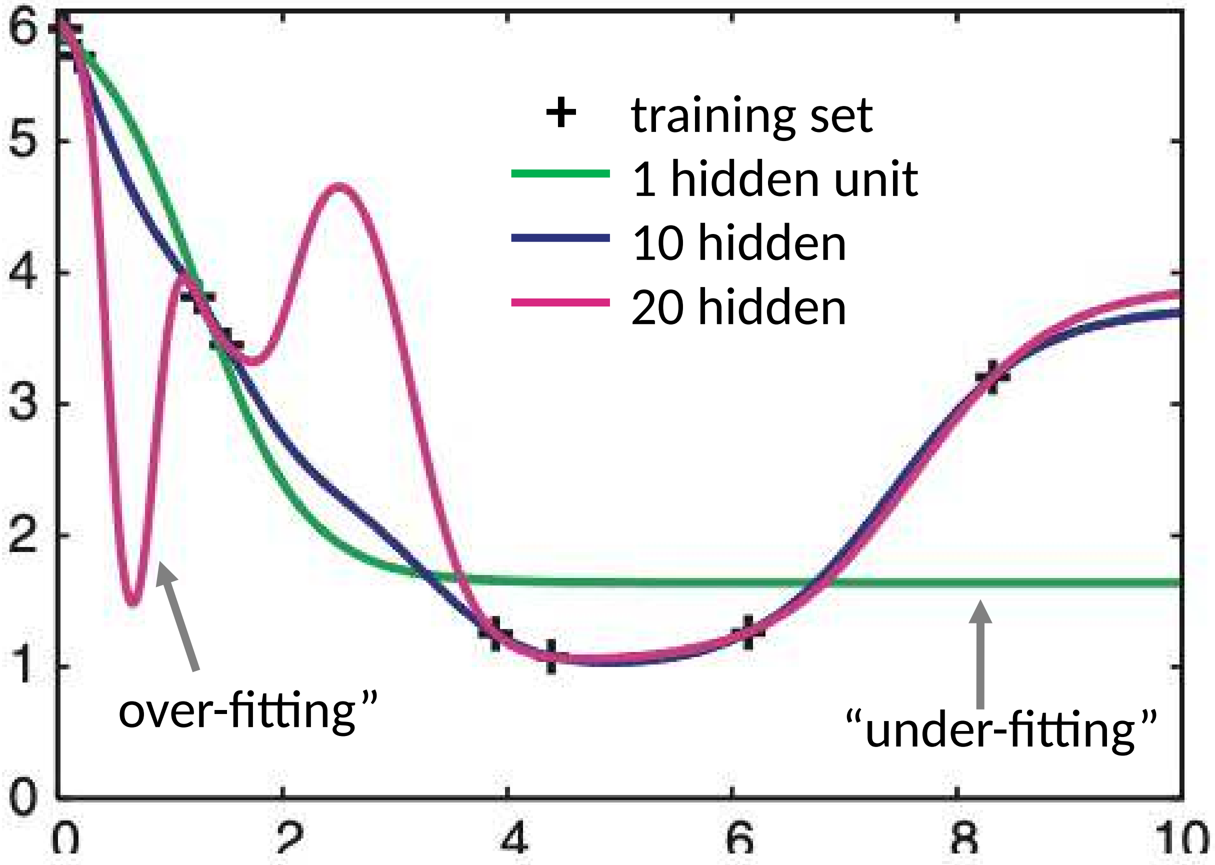
University of Copenhagen

Over-fitting and generalization

- Many parameters and few training data leads to over-fitting

- If it over-fits, the network cannot generalize

- To generalize means to be able to predict on unseen (test) data



From A Krogh (2008) Nat. Biotech. 26, p. 195

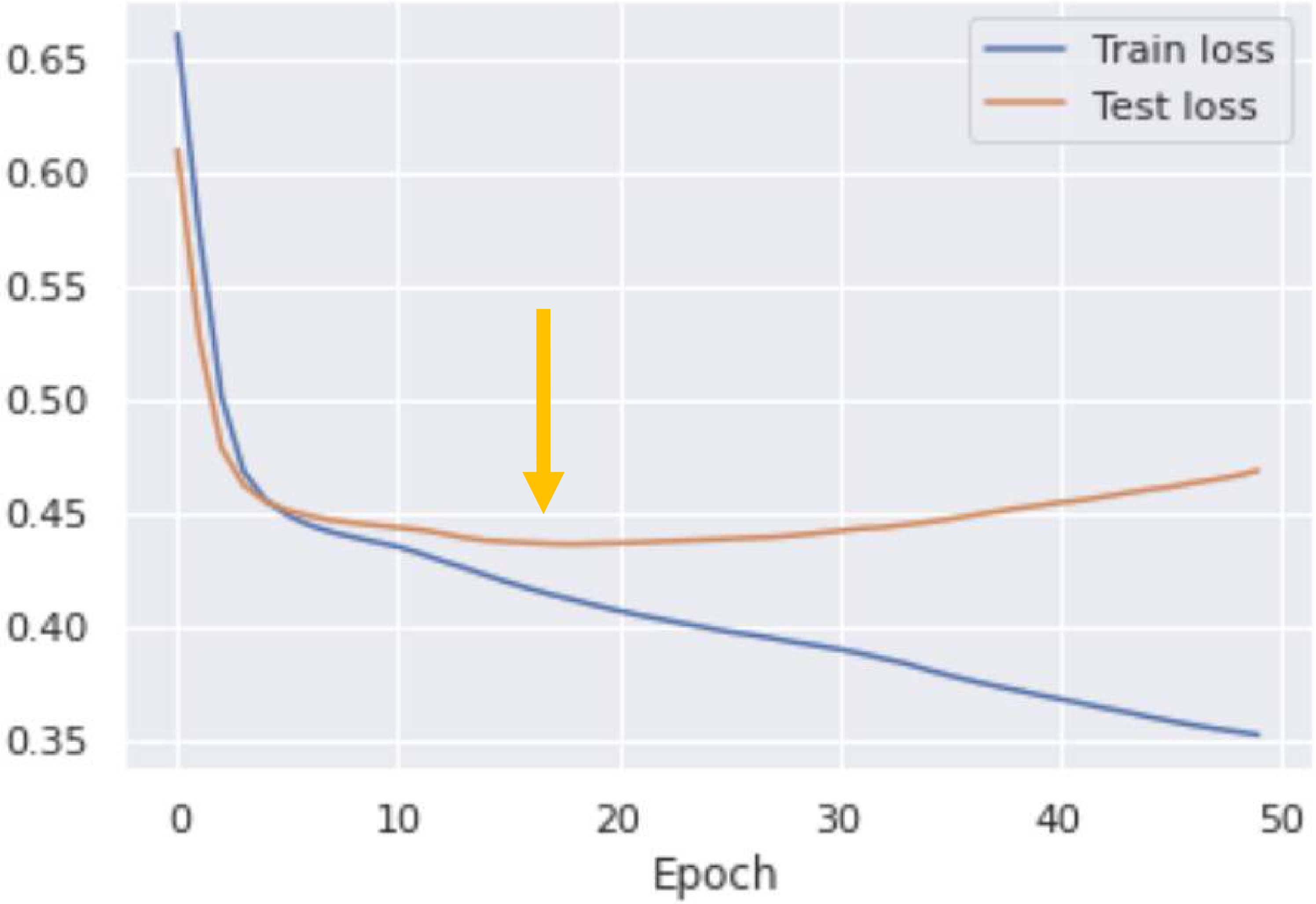
Over-fitting

Sign of over-fitting:

Test error starts to grow while training error decreases

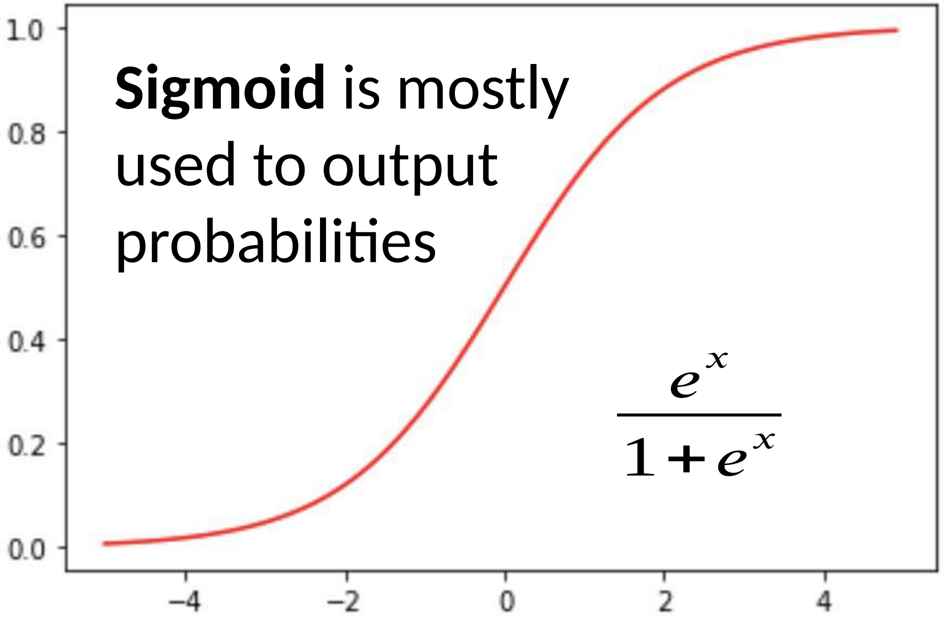
The network size can be decreased if it over-fits (e.g. fewer hidden units)

Alternatively, a weight decay can mitigate over-fitting



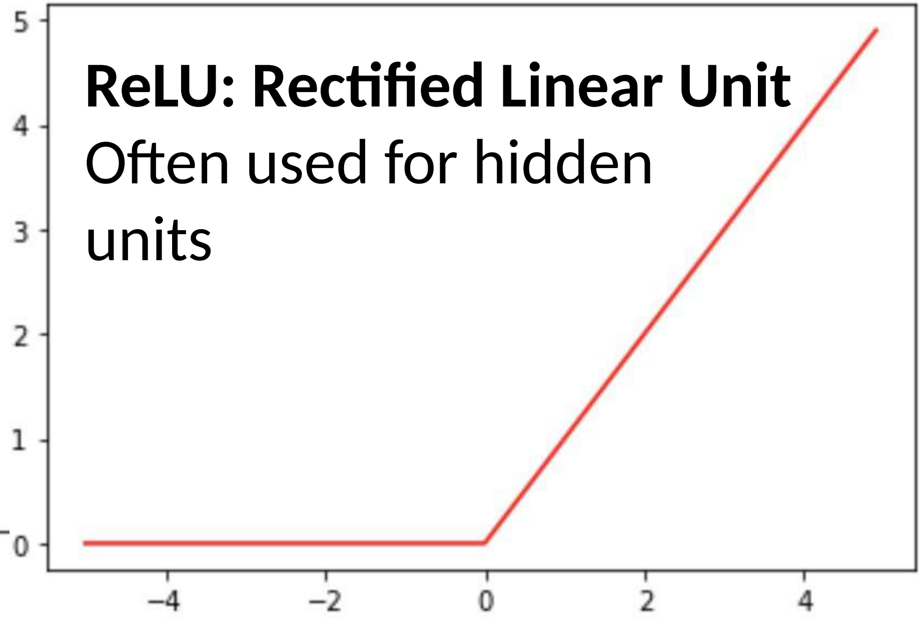
Weight decay: a term is subtracted from a weight in each iteration. is normally small, 10^{-2} to 10^{-6}

Activation functions



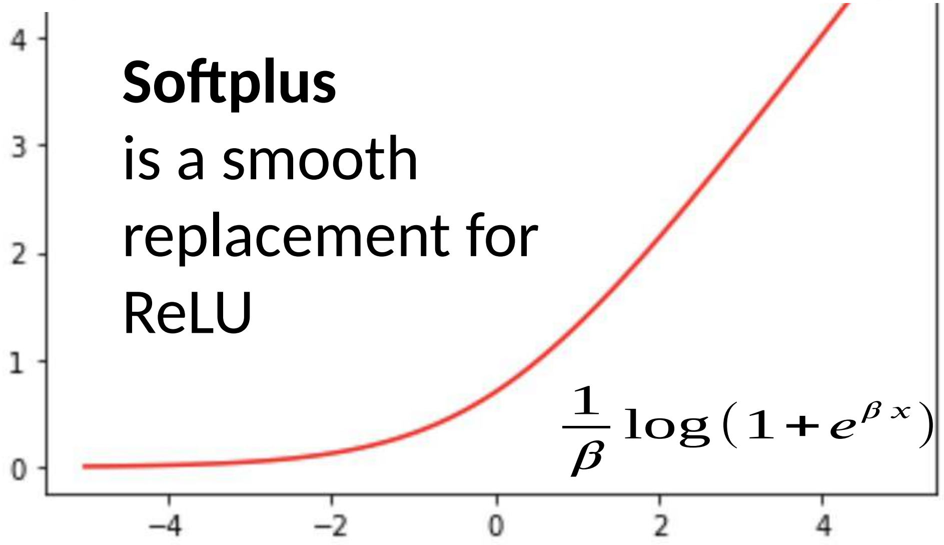
Sigmoid is mostly used to output probabilities

$$\frac{e^x}{1+e^x}$$



ReLU: Rectified Linear Unit

Often used for hidden units



Softplus

is a smooth replacement for ReLU

$$\frac{1}{\beta}\log(1+e^{\beta x})$$

softmax & more on maximum likelihood

Choice of optimizer, parameters, etc

- In stochastic gradient descent (torch.optim.SGD) you need to set parameters (learning rate and momentum)
- The Adam optimizer (torch.optim.Adam) is usually a better choice
  - It automativcally adapts the learning rate and momentum in clever ways
  - It is based on SGD and uses mini-batches
  - you can set a weight decay
- There are many things you can vary in a Neural Network.
- It is a good idea to make an initial “grid search” where you systematically test performance by varying
  - the number of hidden layers and their size
  - other parameters one by one
- This is sometimes done on a reduced data set with quite few iterations